NBSIR 88-3089

AN AUTOMATED HIGH-TEMPERATURE GUARDED-HOT-PLATE APPARATUS FOR MEASURING APPARENT THERMAL CONDUCTIVITY

Jerome G. Hust
B. James Filla
James A. Hurley
David R. Smith

National Bureau of Standards
U.S. Department of Commerce
Boulder, Colorado 80303-3328

May 1988

DEPARTMENT OF COMMERCE · UNITED STATES
Stimulating America's Progress
1913-1988

NBSIR 88-3089

AN AUTOMATED HIGH-TEMPERATURE GUARDED-HOT-PLATE APPARATUS FOR MEASURING APPARENT THERMAL CONDUCTIVITY

Jerome G. Hust
B. James Filla
James A. Hurley
David R. Smith

Chemical Engineering Science Division
Center for Chemical Engineering
National Engineering Laboratory
National Bureau of Standards
Boulder, Colorado 80303-3328

May 1988

Funded in part by
Department of Energy
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37830

U.S. DEPARTMENT OF COMMERCE, C. William Verity, Secretary

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

CONTENTS

Abbreviations and Symbols Used in This Report

A	Area of Metered Section
DTC	Differential Thermocouple
DVM	Digital Voltmeter
G	Gap between Main Heater and Inner Guard
GHP	Guarded Hot Plate
I	Electric Current
IG	Inner (Primary) Guard
k	Thermal Conductivity
K	PID Controller Gain
K_P	Proportional Gain Function
K_I	Integral Gain Function
K_D	Derivative Gain Function
K_{P1}, K_{P2}	Proportional Gain Parameters
MH	Main Heater
PRT	Platinum Resistance Thermometer
Q	Heater Power
R	Resistance
RTD	Resistive Temperature Detector (Thermometer)
S	Standard Deviation
S_{P1}, S_{P2}	Proportional Gain Bellwidths
S_I	Integral Gain Bellwidth
S_D	Derivative Gain Bellwidth
SRM	Standard Reference Material
t_c	Control-cycle time interval
TC	Thermocouple
T_C	Temperature of Cold Surface of Specimen
T_H	Temperature of Hot Surface of Specimen
ΔT_1, ΔT_2	Temperature Difference Across Each Specimen
τ_I	Integral Time Constant
τ_D	Derivative Time Constant
V	Potential difference
ΔX	Specimen Thickness
ZG	Zero Gradient

List of Figures

List of Tables

AN AUTOMATED HIGH-TEMPERATURE GUARDED-HOT-PLATE APPARATUS FOR MEASURING APPARENT THERMAL CONDUCTIVITY

Jerome G. Hust, B. James Filla,
James A. Hurley, and David R. Smith

Chemical Engineering Science Division
National Bureau of Standards
Boulder, Colorado 80303-3328

An automated guarded-hot-plate apparatus was designed and built to meet the requirements of ASTM Standard Test Method C-177 for measuring the thermal conductance of thermal insulation. Apparent thermal conductivity has been measured with this apparatus in the range from 40 to 100 mW/(m.K) at mean temperatures from 300 to 750 K, in environments of air and helium, at pressures ranging from atmospheric pressure to roughing-pump vacuum. The apparatus is controlled by a desk-top computer. A thermocouple device of novel design more accurately senses the average temperature over the surface of each heater plate. An improved algorithm for the control sequence leads to more stable heater powers and specimen temperatures. Initially it brings the system rapidly to a temperature setpoint with minimal overshoot. It also permits highly sensitive control of the plate temperatures in later phases of the measurement sequence when thermal stability of the specimen boundaries is very important in measuring the thermal conductivity with high precision. This algorithm has enhanced the performance of both the high-temperature and the low-temperature guarded-hot-plate apparatus at this laboratory. The apparatus can be operated at either constant hot-plate temperature or constant heater power.

Overall uncertainties of apparent thermal conductivities at atmospheric pressure are 2% at 300 K and 5% at 750 K when measuring conductivities in the range from 40 to 100 mW/(m.K). The apparatus will be valuable in development of new Standard Reference Materials of low conductivity and for higher temperature ranges, and is being used in comparative interlaboratory measurement programs.

Key words: automated control system; guarded-hot-plate apparatus; high temperature; mean-temperature sensor; thermal conductivity; thermal insulation; thermocouple device.

1. INTRODUCTION

The U.S. National Bureau of Standards (**NBS**) establishes Standard Reference Materials (**SRMs**) needed to improve accuracy in measurement of physical properties. Within NBS the Center for Chemical Engineering (**CCE**) has helped to develop SRMs of thermal resistance for over twenty years. During the past ten years CCE helped establish SRMs for thermal insulation at temperatures from 100 K to 330 K. As a result of this effort CCE established two insulation SRMs, fibrous glass board SRM 1450b [**1-5**], and fibrous glass blanket SRM 1451[**6**], in cooperation with the NBS Center for Building Technology. The low-temperature data for certifying these SRMs from CCE were obtained with a guarded-hot-plate apparatus designed to be used at ambient and low temperatures. This apparatus has been described by Smith, Hust, and Van Poolen [**7**].

Both industry and NBS have for several years recognized the need for thermal insulation SRMs for use at higher temperatures, but limited funds and manpower delayed the start of this effort until insulation SRMs for the building industry were completed. Several years ago CCE began to design and construct a guarded-hot-plate apparatus capable of measuring materials having apparent thermal conductivities in the range of about 20 to 200 mW/(m·K) and for temperatures ranging from room temperature to about 750 K.

This apparatus has been completed and tested to establish its precision and bias. The tests included determining the thermal conductivity [**8,9**] of SRM 1450b fibrous glass insulation board at temperatures from 300 to 350 K and measurement of two candidate insulation SRMs for high temperatures. These measurements use double-sided heat flow as well as single-sided [**10**] heat flow; both small and large temperature gradients; runs with temperature offsets on the primary and secondary guards; environments of different gases such as ambient air, dry air, dry nitrogen, argon or helium; and investigations of the effect of drift of the specimen temperature. The results of these measurements, along with comparisons to previously published data, are used to assess the precision and bias of the new apparatus.

Ideally a guarded-hot-plate (**GHP**) apparatus should produce conditions of stable and straight-line flow of heat from a source of known power. The entire output of this source should flow through, and along a direction normal to, a known or "metered" area of the surface of a specimen. For practical reasons a matched pair of specimens is used, sandwiching the thermally guarded, main heater plate, and equally sharing the power. Stably controlled, cooled surfaces sink the heat energy. Measurement of the temperature gradient along the direction of heat flow and within the metered area of the specimen then allows one to compute the thermal conductivity as the ratio of heat flux (time rate of heat flow per unit area) to temperature gradient. Additional heater plates permit changing the mean temperature of the specimen independently of the temperature gradient, which is established by the main heater plate. The collection of heater plates, specimens

and cooling plates is called the "stack".

The major task of the designer of a GHP apparatus is to approach as closely as possible the ideal conditions described above. Measuring the true gradient within the specimen, ensuring straight-line flow of heat, and avoiding the loss of heat produced by the metered heater or the introduction of extraneous heat flow through the metered area of the specimen, can be particularly challenging. When measuring rate of heat flow through a material designed to be a good insulator, it is obviously difficult to insulate such a specimen to prevent loss of heat! Establishing within the specimen a mean temperature that is close to the ambient temperature will minimize errors in measuring the rate of heat flow through the specimen. However, contact resistance between the heater plates and the specimens is a source of error when measuring specimens of relatively high conductance. While we may consider avoiding this problem by mounting the thermocouples or other temperature sensors directly on the surface of the specimen, it is often desirable not to change the surface of the specimen by making grooves for the sensors or introducing adhesive material to hold the sensors.

The initial design of the high-temperature GHP apparatus was guided by the following criteria: (a) the size of the hot plate should be typical of those commonly in use, (b) the temperature range of the apparatus should be large enough to satisfy a significant need but not so large as to delay its timely completion, and (c) the apparatus should be totally automated both to control the measurement process and to acquire the data. A logical extension of item (c) was to construct the system for control and data acquisition in such a way that it could be used to operate the previously described low-temperature GHP system. The low-temperature apparatus was rewired to exploit the advantages of the automated control apparatus. It has been operated successfully to measure thermal insulation SRM 1450b [3], having a thermal conductivity of 40 to 60 mW/(m•K), while checking out the proper operation of the automatic control system.

2. DESCRIPTION OF APPARATUS ELEMENTS

This GHP apparatus is consistent with the specifications of ASTM Standard Test Method for "Steady-State Heat Flux Measurements and Thermal Transmission Properties by Means of the Guarded-Hot-Plate Apparatus" (C 177-85) [8]. It differs from the low-temperature apparatus [7] only in the materials used in its construction and in the diameter of its measurement stack. The stack elements of the GHP apparatus and the associated environmental control components are illustrated in figure 1. Figures 2 and 3 show block diagrams of the electronic system for control and data acquisition. For those who may wish to take advantage of the developed program, written in BASIC language, Appendix A contains a listing of the entire computer software package that completes the automation of the system.

2.1 Guarded-Hot-Plate Stack and Environmental Control Elements

The stack, shown in figure 1, consists of the following elements from the bottom to the top: bottom cold plate, bottom heater offset insulation, bottom auxiliary heater plate, bottom specimen, main and inner guard heater plate assembly, top specimen, top auxiliary heater plate, top heater offset insulation, and top cold plate. The inner guard heater plate, surrounding the main heater plate, is also referred to as the primary guard. A cylindrical outer, or secondary, guard surrounds the stack. Alumina, with its high melting point and relatively high thermal conductivity for an insulating material, was chosen as the material for the outer guard and heater plates. The 25-mm thick sintered-alumina heater plates, having masses of 3.2 kg, and the 6-mm thick copper cold plates, are circular disks 25.4 cm (10.0 in) in diameter. A 25-mm thickness of calcium silicate heater-offset insulation permits the use of a wide range of specimen temperature differences within the limitations of the power supplies.

The heater plates and outer guard were fitted with the necessary sensors to measure and control their temperatures. Nicrosil-Nisil* thermocouples, resistant to the effects of high temperatures, are used to sense the temperatures of the heater surfaces adjacent to the specimen surfaces and to control the temperature difference across the gap between the metered region and the primary guard. The sensitivity of this alloy increases from 28 to 38 μV/K over the range from 300 to 800 K. Sensitive control of each heater plate and each outer guard is achieved by mounting platinum resistance thermometers (PRTs), encased in a refractory material, on the heater assembly. Each of the wire leads for the heaters and temperature sensors was thermally anchored to its associated plate and was wrapped around and cemented to a thermal anchoring post in the midplane of the vertical outer surface of the heater plates. Each wire was then wrapped into a coil about 3 mm in diameter between the edge of the plate and the base plate of the stack. The base plate is water-cooled to keep it near room temperature.

---

*Trade name is given here purely for identification purposes and does not imply endorsement by NBS. Similar products of other manufacturers may work as well or better.

Each cold plate comprises two copper disks soldered together. Machined next to each other in the inner surface of each disk are two symmetric spiral grooves, connected at the center. These form a counterflow heat exchanger for the entering and exiting streams of coolant, and minimize the temperature differences within the cold plates induced by the temperature difference between the two streams. The coolant lines were connected to the top and bottom cold plates in a parallel arrangement so that the rate of flow can be controlled by a separate valve for each plate. Water from the domestic supply is the coolant, but any other suitable coolant could be used, such as chilled alcohol from a refrigerated bath.

Each stack heater plate (top, main and bottom) was constructed from two alumina disks, one of which was grooved. The grooves were cast as two adjacent spirals connected at the center of the plate. This design is a convenient way to distribute the heating power uniformly over the heater area. It also minimizes the area enclosed between the two leads, which minimizes any inductive coupling to nearby sensor circuits in case it is desired to power the heater with ac power. Platinum heater wire was held within the grooves with refractory cement, which also holds the two plates together. The thicknesses of the disks and depth of the grooves are such that the plane of the heater wire lies midway between the top and bottom surfaces of the assembled heater plate.

To make the main heater-inner guard plate, the centers of two plates identical to those used for the auxiliary heaters were cut out, with diameters half of that of the whole plate. The central core became the main heater, and the annulus, the inner guard.

The outer guard is a 4.6 kg circular, cylindrical shell of cast alumina 6 mm in thickness and 20 cm in length surrounding the stack from the level of the bottom auxiliary heater plate to that of the top auxiliary plate. Its inner diameter is 2.5 cm larger than the diameter of the stack plates. Platinum heater wire was wrapped non-inductively on the outside of this guard and cemented in place.

An aluminum shroud surrounds the stack and contains the loose-fill insulation which limits the loss of heat from the stack. The shroud is 46 cm high and 58 cm in diameter. Its position on the water-cooled base plate helps keep it near ambient temperature.

Surrounding the entire stack/shroud assembly is a stainless-steel environmental-control chamber with an O-ring seal on its bottom edge. This chamber allows us to introduce different gases or to evacuate the space around the stack to a vacuum of about 7 Pa (50 μm). The type of loose-fill insulation (exfoliated mica) used around the stack has a large surface-to-volume ratio and is a good moisture adsorbent; this makes it difficult to achieve pressures of less than 10 Pa in a short time.

2.2 Mounting of Thermocouples and PRTs

To sense the absolute temperature needed to obtain the thermal conductivity, with minimum distortion of the isotherms and heat flux lines within the specimens, thermocouples are used at the centers of the heater plate surfaces. To control the desired plate temperatures, four-lead PRTs are used as sensors in the control circuits.

One PRT was cemented into a slot machined in the center of the outer edge of each auxiliary heater plate. Another PRT was similarly mounted on the metered (main) plate within the gap between it and the inner guard. The PRT for the outer guard was cemented to its outside surface.

A unique three-lead thermocouple combination was designed to measure accurately the average temperature over the surface of each heater plate. This three-lead device results in significantly more accurate measurements of the average temperature of a surface compared to those obtained with a single thermocouple. See Appendix B for a more detailed discussion of the theoretical basis, construction and comparative performance of this device.

The temperature difference across the gap between the metered plate and the primary guard is controlled with a twenty-element thermopile. Ten thermopile elements (the first half) lie on the top surface of the plate and the other ten (the second half) are on the bottom. Control is exercised using the whole thermopile but a third lead is connected to the junction between the two halves so that the top and bottom thermopiles can be monitored separately. The junctions of the thermopile are uniformly distributed along the circumference of the gap at locations alternating between its opposite sides, approximately 4 mm from the edges of the gap.

2.3 System for Data Acquisition and Control

The system for acquiring data and controlling temperature and power is composed of two principal parts: (a) the computer (with associated multiprogrammer, interface cards and software) and (b) power supplies and digital voltmeters (DVMs). The schematic arrangement of these components is shown in figures 2 and 3.

The system contains five power supplies for the heaters in the five plates and guards, a single current supply for the five PRTs on the heater units, and six digital voltmeters. Three DVMs respectively read the potential differences across the PRTs sensing the temperatures of the main heater plate and two auxiliary heater plates. A fourth DVM monitors the emf from the gap thermopile used to maintain equality of temperature between the main heater and primary guard. The fifth DVM reads the emfs for the thermocouples sensing various plate temperatures, as well as the output from the PRT used to sense the temperature of the outer guard.

Each thermocouple emf is selected for reading by the computer through the use of two switching-relay modules having low levels of parasitic thermal emfs. Each switching module has ten relays, i.e., ten double-pole switch positions. These relays are also used to determine the currents in the PRTs and main heater by allowing the fifth DVM also to read the potential drop across a standard resistor in each circuit. The power delivered to the main heater is calculated from values of the current in and the potential drop across the main heater resistance. This potential difference is measured with the sixth DVM.

Four of the five power supplies for the heater plates are capable of a maximum output of 55 V and 5 A (275 W). For maximum power transfer the optimum heater resistance is therefore 11 Ω. The actual heater resistances, ignoring lead resistances, are:

(a) main heater, 3.7 Ω;
(b) top and bottom auxiliary heaters, 14.3 Ω;
(c) inner guard, 10.2 Ω; and
(d) outer guard, 11.9 Ω.

The outer guard requires a power supply of at least 75 V to supply the power needed at the highest temperatures of operation. The low resistance of the main heater is not a problem, because it is thermally well shielded against loss of heat and requires only a low power even at high temperatures.

The PRT current supply is manually set to the optimum current as calculated by the computer for each set of runs. The optimum current for the temperature range of this apparatus varies from 0.6 to 1.5 mA. At room temperature the resistance of each PRT is 100 Ω.

Thermocouple emfs and the power to the main heater are measured by two DVMs having a precision of 6 1/2 digits (0.01 μV). The four DVMs used to measure the gap thermopile emf and the PRT voltages on the main heater and two auxiliary heaters have a precision of 5 1/2 digits (0.1 μV). The analog-to-digital conversion card used to measure the outer guard PRT voltage has a precision of 3 1/2 digits. The heater control circuits for all but the main heater provide for changes in power supply voltage as small as 10 mV. The control circuit for the main heater can change its output by values as small as 1 mV. The low-thermal-emf relay modules are designed to provide selector switches free of spurious emfs down to levels of 20 nV.

The core of the control system is a modified digital PID (Proportional, Integral and Derivative) controller of novel design and based in software residing in a scientific personal computer. Its algorithm for control of the power supplied to the heater plates provides for vigorous heating and rapid approach to the desired temperature when the controlled temperature is far from the set point, yet gives very sensitive control when the controlled temperature is near the set point. These features are a result of variable controller gains. The control algorithm is described in detail in Appendix C.

2.4 Specimen Thickness Spacers

Some specimens to be measured compress under the weight of the overlying plates, the thickness of the specimens then varying with time due to creep. To slightly compress these specimens, and thereby assure good thermal contact between them and the adjacent heater plates, the plates are held apart with rigid spacers cut from stainless steel tubes. These spacers have walls 0.25 mm thick and are shorter than the specimen thickness by about 0.2 mm. Three equally-spaced notches are cut into the outer periphery of each specimen to hold the spacers between the outer edges of the inner guard and the auxiliary heaters. Each spacer is cut to the desired length within a tolerance of 0.025 mm. The tubes are filled with refractory fibrous insulation, to reduce the possibility of radiative and convective heat transfer within their interiors. These modes of heat transfer would shunt the conductive heat transfer through the specimens.

Although the tubes thermally connect the outer edges of the inner guard and auxiliary heaters, their presence is necessary to preserve constancy of the thickness of compressible specimens. The shunting effect of the spacer tubes is minimized by their thin walls, fibrous inner packing and poor (point) contact with the ceramic plate surfaces. Their position at the outer margins of the stack minimizes the effect of their shunting on the direction of flow lines of the metered heat through the specimens.

3. OPERATION

After the specimens have been prepared and installed, flow of coolant to the cold plates is initiated. The power supplies, DVMs, multiprogrammer and electronic relay modules are turned on and the computer program started. During the initial part of the run the computer prompts the operator for the time and date, identification and characteristics (mass and thickness) of the specimen, the environmental gas, and run information. This last item includes the values of temperatures to be maintained at the surfaces of the specimens, and whether flow of heat is desired to be single-sided (through either specimen, as selected) or double-sided (through both specimens). The operator may choose whether the system should be controlled at constant temperature or at constant power. Up to nine runs can be programmed at one time, each with unique conditions of operation.

After the operator has keyed in the information, the computer begins automatically controlling the experimental conditions and acquiring data. During this automated sequence, the operator may view on the computer monitor various plots showing the behavior of any of the controlled circuits as a function of time. In addition, the operator may change setpoint temperatures and other control parameters during the run.

Each measurement sequence is divided into three phases. During phase I, the desired temperatures are established at the surfaces of the heater plates facing the specimens, using for temperature control the temperatures sensed by the PRTs. These temperatures are compared to the thermocouple temperatures of the main and auxiliary heater plates during Phase I. If a difference of more than 10 mK exists, the control setpoints are adjusted so that the desired temperatures are obtained as sensed by the thermocouples on the plate surfaces. When these temperatures and the main heater power have stabilized within preset limits, the sequence enters Phase II.

During this second phase thermal conductivity is computed every three minutes. After the first thirty minutes of Phase II the main heater power and the calculated thermal conductivity are examined for stability within specified limits, allowing entry into the third phase if the stability criteria are met.

In Phase III the operator is allowed to plot on a dot-matrix printer a permanent record of the history of the various plate temperatures, the power to the metered heater, or the thermal conductivity calculated as successive 3-minute time averages. Then, if the operator does not intervene the computer automatically averages the last 30 minutes of data for storage on disk. Data from any longer interval may be averaged if the operator so chooses. All of the data from Phase II is saved to disk for later re-analysis as desired.

Table 1 shows the data printed out at the end of a typical experimental sequence, using a specimen of microporous fumed silica insulation board. A summary of the conditions of the experiment is given. Also included are final values of measured temperatures and temperature differences, heater power, and thermal conductivity for the experiment. Statistical measures of the random variations of these quantities, averaged over the interval of stability at the end of the run, are listed. Corrected data for sample thickness and main plate area result from considering thermal expansion of these elements resulting from heating them from room temperature to the temperature of measurement; the main plate area is also adjusted to include half the gap area.

Both Phase I and Phase II have predetermined maximum time limits. The maximum time limit for Phase I is 5 hours and that for Phase II is 10 hours. The combined maximum time limit for the whole experiment is 11 hours. These limits force the run to be completed after a definite time interval in cases where the stability requirements are too stringent in the control software.

Typical graphical outputs that may be viewed during the measurement sequence and printed at the end of the run are given in figures 4 through 18. They illustrate conditions during measurement of (arbitrarily chosen) microporous fumed silica insulation board and correspond to the information given in Table 1. The even-numbered figures 4 through 16 show the behavior of the system during the initial approach to the respective setpoints of the various heater plates. The odd-numbered figures 5 through 17 depict the behavior of the system during final stable operation at the respective setpoints. Figure 18 gives the temperature history of the thermocouple reference block.

The even-numbered figures 4 through 16 illustrate several noteworthy aspects of the operation of the control system using the PID algorithm described in Appendix C. For example, figure 12 shows that the temperature of the main heater was raised by about 110 K in less than 30 min without overshooting the setpoint. Figure 4 shows that during the same interval the output emf of the gap thermocouple went to zero in about 35 min, also without overshoot. (As the emf of this differential thermocouple went to zero, the inner guard temperature caught up with that of the main heater.) The temperatures of the top and bottom auxiliary heater plates took a little longer to reach their setpoints, about 55 min (figs. 8 and 10).

Figure 7 reveals that a less-sensitive level of control was used for the outer guard: the fluctuations in its temperature are about ±0.3 K. This is quite acceptable since the temperature of the outer guard is only relatively loosely coupled to that of the measurement stack.

Figure 14 shows that the power to the main heater plate initially called for by the control algorithm was about 60 W for the first 25 min, and then dropped rapidly down to about 4 W. Thereafter the power decreased gradually to the final stable value of 2.4 W (fig. 15). This represents a range of about one and a half orders of magnitude of power. In about 60 min a value of heater power very near the final stable value is reached (a considerably longer time is of course needed for the whole experimental stack to come to its stable state). It is also noteworthy that the power fluctuation amplitude gradually decayed to relatively small values ($\pm$ 0.2 W) near the end of the trace in figure 15. This behavior is a direct result of the PID control algorithm, when operating in the constant-temperature mode.

Figure 13 depicts typical behavior of the final temperature of the main heater plate during stable operation at the setpoint. The amplitude of the temperature fluctuations is about $\pm$ 5 mK.

According to figure 16, the experimental thermal conductivity is determinable to within about $\pm$ 10% after about 35 min. This is cited not to claim any ability to determine thermal conductivity accurately within this short an interval of time, but rather to illustrate how rapidly the control algorithm brings the system to the setpoints desired.

The data illustrated in figure 17, averaged every three minutes over the last sixty minutes of the run, gave a value for thermal conductivity of 22.07 mW/(m.K), with a standard deviation of 0.028 mW/(m.K). This represents a relative deviation of 0.12%, which is typical behavior for most temperatures of operation.

4. PRIMARY MEASUREMENTS

The measurements required for the computation of average thermal conductivity k at a temperature T are governed by the defining equation,

$$k(T) = Q\,\Delta X/(A\,\Delta T), \qquad (1)$$

where Q is the rate of flow of heat through the metered portion of a single specimen, ΔX is the average thickness of the specimen, A is the metered area and ΔT is the mean temperature difference across the specimen. The value of T associated with the computed value of k must be carefully defined, as will be discussed below.

The normal mode of operation of the GHP apparatus involves flow of heat through two sides of the main heater plate through two matched specimens. This is referred to as the double-sided mode and results in an average value of k for the two specimens. We can also adjust the temperature of one of the auxiliary heaters to match closely that of the main heater, thus minimizing the flow of heat through the specimen in that direction. This is referred

to as the one-sided mode [10] of operation and results in the k value for the other specimen, through which most of the heat flows. To accommodate both modes of operation, the following equation is used to calculate thermal conductivity:

$$k = Q\,\Delta X/(A(\Delta T_1 + \Delta T_2)), \qquad (2)$$

where Q is the total power generated by the metered heater, flowing through both specimens, and ΔT_1 and ΔT_2 are respectively the temperature differences across the two specimens. In the special case where either ΔT_i is zero, this reduces to eq(1). When both ΔT's are equal we obtain the same expression as that for the case of one specimen of area 2A, conducting a power Q.

For the one-sided mode either ΔT_1 or ΔT_2 is small but this small value can be either positive or negative. The specimen with the smaller ΔT is sometimes referred to as the "back" specimen because its purpose is to minimize the flow of heat from the main heater plate in the "back" direction, away from the specimen being measured.

The term "average thermal conductivity" has been used here to denote two types of averaging. First, in the double-sided mode the arithmetic average value of k for the two specimens is obtained. No information on either specimen individually can be calculated from measurements in this mode. Second, the value of k averaged with respect to temperature from the cold-face temperature, T_C, to the hot-face temperature, T_H, is obtained in either the single- or double-sided mode. As a rule of thumb, if this temperature difference ΔT between the two faces is less than 10% of the absolute mean temperature of the specimen, $(T_C+T_H)/2$, the value of k can be assigned to the mean temperature with an error of less than 0.1%. This is immeasurable, given the current state of the art in measuring thermal conductivity. If ΔT is significantly larger, the k value obtained may be measurably different from the true k value corresponding to that mean temperature. In such cases, the integral method of analysis as described by Hust and Lankford [11] is used to obtain k as a function of temperature.

There is also a third type of averaging involved in the calculation of k using this apparatus, i.e., an average over time. This is done to reduce the effect of the imprecision of the individual readings. The minimum time interval for this averaging process, occurring in Phase II, has been set to 30 min. During this period values of temperature are obtained every 3 min; thus the time-averaged value of T involves a minimum of 10 readings. The values of potential difference and current needed to obtain the power, Q, are read every 5 s, so the time-averaged value of power involves a minimum of 360 readings.

5. ANALYSIS OF PRECISION AND BIAS

The apparatus is designed to determine heat flow through the specimens under a prescribed set of boundary conditions; from the heat flow one can subsequently calculate the apparent thermal conductivity, k, when applicable. The analysis of errors (uncertainties) will be directed toward determining the imprecision and systematic uncertainty (bias) of k. Errors due to the inapplicability of the definition of k for a particular material and to inhomogeneity within the material will be ignored. Only uncertainties in experimental control and measurement will be considered.

The imprecision and bias of this apparatus were analyzed by two methods: (a) from a propagation-of-error analysis of the imprecision and bias inherent in the measurement of each primary variable, and (b) from the measurement of standard reference specimens. The propagation-of-error analysis requires detailed knowledge of the potential errors of each measuring instrument as well as knowledge of any existing deviations from unidirectional heat flow and from stability over time. It requires a vast amount of experimental effort to produce estimates of imprecision and bias that are reliable. The measurement of standard reference specimens for estimating imprecision and bias is easier, but for temperatures above 350 K, no standard reference specimens of thermal insulation exist at the present time.

5.1 Analysis of Propagation of Errors

The error-propagation formula for the defining equation takes the form

$$S_k{}^2 = \left(\frac{\partial k}{\partial T_M} S_{TM}\right)^2 + \left(\frac{\partial k}{\partial Q} S_Q\right)^2 + \left(\frac{\partial k}{\partial A} S_A\right)^2 + \left(\frac{\partial k}{\partial DX} S_{DX}\right)^2 + \left(\frac{\partial k}{\partial DT} S_{DT}\right)^2. \quad (3)$$

In this relation, S is a standard deviation, the subscript k represents thermal conductivity, T_M is mean specimen temperature, Q is heater power, A is area of the metered (main) heater, DX is the specimen thickness, and DT is the temperature difference through the specimen. The heater power Q is obtained from the product of potential difference and current through the main heater. Strictly speaking only normally distributed errors (including imprecisions) propagate according to this formula but the errors due to biases will be propagated using the same formula for simplicity. That is, each term of the form of S^2 in eq(3) will represent either $S^2{}_{IMP}$ or $S^2{}_{BIAS}$. The imprecisions and biases of the primary variables are estimated using 95% confidence intervals (two standard deviations) in the sections below. These estimates and the resulting estimated imprecision and bias for k are summarized in Table 2.

5.1.1 Rate of Heat Flow

The heat power Q through the specimen(s) is equal under conditions of stable and unidirectional flow to the power $P = V_H I_H$ supplied by the heater to the metered area. The potential difference across the main heater is V_H and the main heater current is I_H. The uncertainty in Q is the most difficult uncertainty to estimate of all those for factors determining k. This difficulty arises not so much through inaccurate measurement of V_H and I_H as through establishing a flow of heat through the specimens that is both stable and unidirectional along the longitudinal axis of the stack.

The value of V_H is measured directly but I_H is obtained from a measured potential difference V_{STD} across a standard resistance R_{STD} in series with, and carrying the same current as, the main heater resistance: $I_H = V_{STD}/R_{STD}$. Thus the uncertainty in Q depends on the uncertainty in the values of V_H, V_{STD} and R_{STD}.

The uncertainty in the value R_{STD} of the standard resistance is stated in its certificate of calibration and is of the order of parts per million. The two values V_H and V_{STD} have negligible instrumental bias because the DVM's used to measure them are calibrated. The instrumental imprecision of these potential differences is estimated as 10^{-5}, or 0.001%, when measuring power, values of potential typically ranging from 0.1 to 6 V. For thermocouple emfs of 10 μV the imprecision is 0.2%.

One possible systematic error in V_H depends on where the potential leads are attached to the main heater resistance; for the corresponding potential difference to be correct the heat flowing through the metered area must be precisely the energy dissipated in that portion of the heater resistance wire between the potential taps. If for example the potential leads were attached too close to the center of the metered resistance wire then more heat would be dissipated within the metered area than would be calculated from the measured potential difference between the leads. This error (bias) is estimated to be less than 0.13%.

In practice a far more serious error in determining Q lies in the assumption of unidirectional heat flow from the main heater through the metered areas of the specimens. If the metered area were surrounded by a guard area at exactly the same temperature then no heat would be lost from the metered area to the guarded area, along the radial direction. The flow of heat would be unidirectional, and along the direction of the symmetry axis of the stack (axial).

Obtaining this condition is attempted by controlling the heater power for the metered area with the output of the differential thermocouple (DTC) bridging the gap between the metered area and the primary guard, as described earlier in Sect. 2.1. This DTC contains twenty pairs of junctions to increase its sensitivity to temperature imbalances. The control circuitry

attempts to regulate the guard heater power so that the output of the DTC is zero. In practice this output randomly deviates from zero with an imprecision of about $\pm$ 2 mK.

An estimate of the effective thermal conductance across the gap yields a value of 85 mW/K of imbalance due to conduction through air in the gap. An estimate of the effective radiant heat transfer gives a value of about 40 mW/K of imbalance at room temperature, and a worst-case value of about 600 mW/K due to radiation (at a mean temperature of 750 K, the highest temperature usable in this apparatus). Convection is made negligible by the use of mineral fiber insulation in the gap. Conduction through the thermocouple wires bridging the gap is also negligible due to their resistivity and small diameter.

For a total value of about 125 mW/K of imbalance at room temperature, and a value of 690 mW/K at 750 K, the random temperature fluctuation of $\pm$ 2 mK is found to lead to random heat power fluctuations across the gap of 1.4 mW at 750 K. For these worst-case conditions of high temperatures and high radiative transfer across the gap as assumed here, powers are at least 2.5 W even for very good insulators, so these random fluctuations are less than 0.1% of the heater power. This represents an imprecision of 0.1% in the apparent thermal conductivity.

There is some evidence for a systematic unbalance of temperature between different points on the main heater, based on information (Table 1) on the emfs from the "top" and "bottom" halves of the control thermopile, each half containing ten elements. These emfs often differ by about 1.0 μV (0.004 K) when operating at temperatures near 300 K and by about 25 μV (0.07 K) when operating at 750 K. The temperature differences across the gap can be assumed to be nearly equal to these temperature differences, 0.004 K and 0.07 K. These would indicate, from the sensitivities of 125 mW/K at 300 K and of 690 mW/K at 750 K, imbalances in the heater power of 500 mW at room temperature, and 50 mW at 750 K, due to loss across the gap.

Typical metered powers are at least 1 W at room temperature and 2.5 W at 750 K; the estimated systematic error in power could then be 0.05% at room temperature and 2% under worst-case conditions at 750 K. The typical systematic error in power at 750 K is estimated to be 1.5%.

From data such as those shown in Table 1, under dynamic control the imprecision in power Q is found to be typically 0.2%, and seldom worse than 0.5%, at room temperature. At 750 K the imprecision in power rises to typically about 0.5% and is seldom worse than 0.7%.

5.1.2 Thickness and Area of Specimen

For a given assembly of the stack, the thickness and metered area of the specimen do not vary with time except as they are affected by thermal expansion. Compression of the specimen is limited by the use of spacers. For the conditions of thermal stability under which data are obtained, any thermal expansion taking place within the stack is negligible. Thus the imprecisions of the thickness and of the area are taken to be negligible.

The bias, however, is affected by length measurements performed at room temperature and by the degree of validity of the thermal expansion corrections. Changes in spacer lengths and in the diameter of the metered area are allowed for as the stack is heated to the temperature of measurement. The specimen thickness is estimated to be correct to within 0.08 mm as determined by the spacer thicknesses, whose lengths are nominally 25 mm. This yields a relative error (bias) of 0.3%. The area of the specimen, including the gap correction, is estimated to be correct to within 1 cm^2. For a total metered area of 128.9 cm^2 this represents a relative error of 0.8%.

5.1.3 Mean Temperature of, and Temperature Difference Across the Specimen

The precision and bias of the temperature difference across each specimen is affected somewhat by instrumentation read-out errors and calibration errors, but primarily by effectiveness of thermal anchoring of the thermocouples, uncertainty in the location and orientation of the effective plane containing the thermocouple beads, and deviations from planarity of the isotherms within the specimen, or equivalently, deviations from rectilinearity of the heat flux lines. Dynamically the imprecision in temperature is affected by the quality of control maintained by the control system.

The domestic water supply is varies by about 0.7 K during the course of a given experiment, and causes fluctuations of about $\pm$ 10 mK at the surface of the specimen. This is acceptable.

The instrumentation bias is negligible due to calibration of the instruments. It is estimated to be less than 0.01% of the temperature. Since the thermocouple calibration is based on an equation the calibration imprecision is negligible. Calibration bias is estimated to be less than 0.04%. Errors in the temperature difference caused by inadequate thermal anchoring of the thermocouples are difficult to assess analytically. This bias is difficult to separate from errors caused by heat loss as described in the next section. The bias caused by uncertainty of the thermocouple measurement plane is estimated to be 0.6% of the temperature difference. Thus the total bias in mean temperature, and also of the temperature difference through the specimen,

summed in quadrature, is estimated to be 0.85%. From data such as that shown in Table 1, summarizing the experimentally observed data and their statistical variation under dynamic control over the time used to calculate the thermal conductivity, the imprecisions due to dynamic variations in T and in ΔT are found to be 0.02%. Combined with the imprecision in reading the TC emfs, this yields 0.1%.

Combining in quadrature the dynamic imprecisions in temperature, power, thickness, area (the last two having negligible imprecision) and temperature difference, we find the imprecision in thermal conductivity to be 1% at 300 K and 5% at 750 K.

5.2 Error Analysis from Experimental Measurements

Several different experimental comparisons have been made to assess the experimental imprecision and bias of this high-temperature GHP apparatus.

5.2.1 Experimental Reproducibility

Thermal conductivities were repeatedly measured for a clay-bonded fibrous alumina-silica thermal insulation board under consideration for adoption as a high-temperature SRM. The imprecision of these results with removal and re-installation of the specimens in new orientations is 0.5% at 300 K.

5.2.2 Measurements on a Fibrous Glass Insulation SRM

One series of comparison measurements of thermal conductivity was made on a pair of specimens of SRM 1450b fibrous glass insulation board from a lot certified in 1982 [3]. This thermal resistance SRM has a nominal thickness of 2.54 cm and a density of 137 kg/m^3. This SRM was established on the basis of measurements made here in 1980 using the earlier low-temperature guarded-hot-plate apparatus as well as those made at ambient temperature by NBS in Gaithersburg [1,2,4,5]. The experimental data obtained for thermal conductivity as a function of temperature was compared with values obtained from the polynomial published in the SRM certification document [3].

Figure 19 shows thermal conductivity data obtained with the high-temperature guarded-hot-plate apparatus and at temperatures from 300 to 345 K, compared with the functional dependence from the fibrous glass SRM certificate. The deviation plot for these data, shown in figure 20, reveals a slight systematic bias between the data points and the thermal conductivity function, but the agreement between data and function is better than 1.8%, about the same as the imprecision projected from the above error-propagation analysis.

5.2.3 Experiments with Offsets in Temperature of Guard Heaters

Experiments were performed in which the temperature of either the outer guard or the inner guard was deliberately controlled at a setpoint different from the normal one. This condition of operation is called a guard offset. It allows us to assess the effects on measured thermal conductivity of small random or systematic deviations of a guard from its setpoint, which may occur during a normal experiment.

5.2.3.1 Outer-Guard Offsets

In the course of experiments on fibrous alumina-silica insulation board, the outer guard was controlled at offsets of $\pm$ 20 K while operating at a mean specimen temperature of 573 K. The power supplied to the main heater, which is proportional to the thermal conductivity, is plotted in figure 21 for each outer-guard offset.

When the outer guard is lower in temperature than the mean specimen temperature (-20 K offset), under stable control the main heater is required to supply slightly more power in response to the resulting loss from the stack, and conversely. For this effect the sensitivity coefficient is -0.02 mW/K. Thus if the outer guard temperature were not equal to the mean specimen temperature, the error in metered power passing through the specimens would be 0.02 mW/K of offset. This offset would produce an error of 0.05% in the value of k for a 1 K offset when measuring a low-conductivity insulation having a (typical) k-value of 40 mW.

For insulation materials of higher conductivity, requiring higher heating power to maintain the desired gradient, the error due to such an outer guard offset would be less at the same mean temperature. In this case the temperature distribution within the elements of the stack would still be the same, and so conductive losses from the stack to the surroundings would be unchanged, while being a smaller fraction of the required increased power.

Figure 7 shows that the random fluctuations in temperature of the outer guard under stable control are $\pm$ 0.4 K. If the main heater could follow the outer guard temperature as rapidly as the fluctuations occur, the resulting power fluctuation would be $\pm$ 0.01 mW. The heat capacity of the heater plate however does not allow it to fluctuate at the frequency observed in figure 7.

We believe that any outer-guard offsets occurring during normal operation at temperatures near room temperature are much less than the deliberately produced offsets. However, at temperatures near the upper end of the range of operation of this apparatus, it is possible that there could be offsets in the temperature of the outer guard of about 10 K. This is based on the observation of differences between the temperature of the

controlling PRT and the measuring thermocouple of the top and bottom auxiliary heater plates, when operating at 750 K.

5.2.3.2 Inner-Guard Offsets

During experiments on fibrous alumina-silica, and at a mean specimen temperature of 573 K, offsets were made to occur in the output emf of the gap thermocouple between the main heater plate and the inner-guard heater (inner-guard offset). This was done by altering the control software so as to shift the inner-guard setpoint from its usual value of 0 μV. The power supplied to the main heater for each of the two inner-guard offsets is plotted in figure 22, along with the zero-offset value.

Consider the case when the inner guard has a lower temperature than the main heater (-40 μV offset). Then under stable control the main, metered heater is required to supply slightly more power in response to its loss to the guard. Conversely, if the inner guard is higher in temperature than the main heater, the main heater will consume less power as it gains power from the guard. The sensitivity coefficient for this effect is -0.6 mW/μV, for the twenty-junction differential thermocouple used, having a twenty-fold amplified sensitivity. Thus the sensitivity per junction is -0.03 mW/μV, equivalent to 0.9 mW/K. If the inner guard tempera-ture were not equal to the temperature of the main heater, the error in metered power passing through the specimens would be 0.9 mW/K of offset in the differential control thermocouple. This offset would produce an error of 2% in the value of k for a one kelvin offset, but only 0.1% in the value of k for a one microvolt offset in the emf of the control thermocouple, when measuring a low-conductivity insulation having a (typical) k-value of 40 mW.

From figure 5 it is noted that the random fluctuations in temperature of the inner guard under stable control are $\pm$1.5 μV. If the main heater could follow the inner guard temperature as rapidly as the fluctuations occur, the resulting power fluctuation would be $\pm$0.05 mW. This fluctuation amplitude, if realized, would be equivalent to a fluctuation of $\pm$ 0.1% in the value of the thermal conductivity. The heat capacity of the main heater plate, however, does not allow it to fluctuate at the frequency observed in Figure 6.

We think that any inner-guard offsets occurring during normal operation at temperatures near room temperature are much less than the above deliberately produced offsets.

No information on the acually existing offsets is available at present. The temperature difference in the specimen must be greater than or equal to 10% of the absolute temperature to avoid other instrumental errors. Thus the effect of errors in metered power due to biases in controlling the inner guard is believed to be negligible, for operation at room temperature, and of the order of 1% at 750 K.

5.2.4 Zero-Gradient Measurements

Inspection of the defining equation (1) for thermal conductivity shows that k is determined from the ratio of Q to ΔT. A 1% error in either Q or ΔT would propagate a 1% error into k. Using only one kind of specimen (fixed thermal conductivity), it is impossible to separate experimental errors in measuring Q from errors in measuring ΔT by examining the experimental thermal conductivity. In order to judge whether the present apparatus suffers predominantly from experimental errors in Q, or in ΔT, or in both, "zero-gradient" measurements on different pairs of specimens of widely different thermal conductivity can however be very informative.

These zero-gradient measurements are made with the control system programmed to keep the two opposite surfaces of the specimens at the same temperature. Under dynamic control the thermal gradient in the specimen under these operating conditions is not exactly zero, so the system may call for a small amount of power to the main heater. From conventional measurements performed separately, the thermal conductivity of the specimen is known as a function of temperature. This knowledge allows us to compute the amount of heat required to flow by thermal conduction from the main heater through the specimens at the given gradient and mean temperature for the conditions of operation.

We could then assume that any difference between this heat and that actually supplied by the main heater is due to errors in determining the true temperature difference in the specimens. On the other hand, we can assume that any difference between this heat and that supplied by the main heater is a measure of the heat loss from the main heater due to experimental error in meeting the conditions of stable, unidirectional flow of heat along the longitudinal axis. It could possibly be due to conduction of heat from the metered heater out of the stack along thermocouple or heater leads, for example. This difference is called the zero-gradient heat.

Figure 23 shows the zero-gradient heat for specimens of fibrous glass SRM insulation board at temperatures from 293 K to 343 K. The line was fitted to the points by a linear least-squares routine.

The intersection of the line with the abscissa at T = 290 K was forced, for the following reason. The electrical leads into the thermal conductivity stack lie near a large baseplate which is water-cooled to about 290 K and cools the leads. Thus 290 K is a "best estimate" of the temperature of the leads, and is consistent with the behavior shown by the data points. At the conditions of zero gradient, there should be no exchange of heat between the stack and the surroundings for a stack temperature of about 290 K. For higher stack temperatures it should be necessary to supply

heat to the stack. This is illustrated by figure 23. The slope of the fitted line is found to be b = 97 mW/K.

Figure 24 shows the zero-gradient heat for specimens of fibrous alumina-silica insulation board, having approximately the same thermal conductivity as the previously mentioned fibrous glass board. The intersection of the fitted line with the temperature axis at T = 290 K was also forced, as explained above. The slope of this fitted line is found to be b = 87 mW/K, in good agreement with that of the data for the fibrous glass board. Thus both materials have approximately the same sensitivity to rise in mean temperature above T = 290 K.

Both figures consistently indicate the presence of a small but measurable loss of heat from the stack, or apparatus error, increasing in direct proportion to the difference between the mean temperature of the stack and the surroundings.

Additional light is shed by measurements in which the temperature difference ΔT is varied, holding the mean temperature constant. Such data were obtained on a pair of specimens of microporous fumed silica insulation board. At a mean specimen temperature of 673 K consistent values for the thermal conductivity could be obtained only for $\Delta T \geq 80$ K. For ΔT less than this value, the experimental thermal conductivity increased roughly hyperbolically as ΔT was reduced. Similar behavior was observed for a mean temperature of 523 K.

The asymptotic value of thermal conductivity at a given mean temperature for large ΔT is independent of ΔT and is therefore the more reliable estimate of the true value. We can calculate the difference between the experimental value of k at low ΔT and the constant value at large ΔT. This is an error in measurement of k for that mean temperature. If we assume that this error is due entirely to error in measuring Q, we can then calculate this value of power corresponding to the error in k, from knowledge of T and ΔT, for each ΔT.

In so doing, for all values of ΔT used we found a roughly constant value of Q = 29 $\pm$ 4 mW for T = 673 K, and a roughly constant value of Q = 6 $\pm$ 2 mW for T = 523 K. Under the assumption that this error in Q is the sole contributor to error in measuring k, we can compare it to the values of zero-gradient heat (Q_{ZG}) from figure 24. This was assumed to be due to loss of heat from the stack by conduction along leads, and therefore independent of specimen conductivity.

Figure 24 shows that at T = 523K, Q_{ZG} is 20 mW, with an uncertainty from the figure of $\pm$ 10mW; extrapolating to T = 673, Q_{ZG} is 33 $\pm$ 10 mW. These values lie in rough agreement, within experimental error, with the values found above from the experiments with decreasing ΔT. Thus both experiments are consistent with the assumption that there is a small heat leak from the stack to the surroundings, that is greater as the mean

stack temperature is increased, and that contributes to error in determining k accurately. Experimentally it was determined that values of T of at least 10% of the absolute mean temperature must be used to minimize the error due to this heat leak.

5.2.5 Comparative Measurements

5.2.5.1 Comparison With Results From Round-Robin Measurement Programs

During the summer of 1987 an interlaboratory comparative measurement program was organized which involved participation by NBS-Boulder and six industrial laboratories [12]. Thermal conductivity was measured for specimens of fibrous alumina-silica insulation board, and for specimens of calcium silicate insulation board, both types having a nominal thickness of 2.5 cm.

For the fibrous alumina-silica insulation the standard deviation of 58 test results from the seven labs was 9.3%. The data from the apparatus described in this report (X's) lay within 7% of the group mean over a range of temperature from 300 K to 770 K (figures 25 and 26). Since the specimens were different, no great weight should be placed on the 7% difference.

For the calcium silicate insulation board the standard deviation of 48 test results from the seven labs was 9.1%. The data from this apparatus lay within 5% of the group mean over the same range of temperature, 300 K to 770 K (figures 27 and 28). Little weight should be given to the 9% difference, for different specimens.

5.2.5.2 Comparison With Measurements at an Industrial Laboratory

In an in-house calibration and standardization program at an industrial research and development laboratory, measurements of thermal conductivity were made on refractory alumina-silica fibrous insulation board similar to that used in the round robin measurement program described above. Data over a range of temperature from 300 K to 1150 K were found to fit the following correlation by Mitchell [13], to within $\pm$ 3.5%:

$$k_a = 0.0142 + 9.167\ T\times10^{-5} + 2.776\ T^3\times10^{-11}\ \mathrm{W/(m.K)}. \quad (4)$$

Figure 29 shows measurements on this apparatus (circles) compared with the above correlation relation (solid line). Figure 30 gives the relative deviation between the experimental data for this apparatus and the correlation equation. There is a systematic difference between the two of approximately 4%.

6. SUMMARY

This report describes the design and performance of an automated guarded-hot-plate apparatus built to meet the requirements of ASTM standard test method C-177 for measuring the thermal conductance of thermal insulation. The apparatus is capable of measuring thermal conductivity over a range of temperatures from ambient up to 750 K.

Important features of the design and construction of this apparatus are the following:

(1) An improved algorithm for the control sequence leading to stable heater powers and specimen temperatures was developed. Initially it rapidly approaches the temperature setpoint with minimal overshoot. It also permits very sensitive control of the plate temperatures in later phases of the measurement sequence when thermal stability of the specimen boundaries is very important in measuring the thermal conductivity with high precision. This high precision of control in turn leads to very good reproducibility of measurements under the same nominal conditions, as has been verified experimentally.

(2) A novel thermocouple design is used which more accurately senses the average temperature over the surface of each heater plate in the apparatus. This design leads to greater accuracy because more points of the surface are sampled. This in turn leads to more accurate control of the absolute temperature of the plate surfaces and of the adjoining surfaces of the specimens. Consequently the measurement of thermal conductivity with this instrument is also more accurate. The arrangement also leads to greater precision in measuring temperature because the design incorporates a thermopile arrangement which multiplies the effect of any temperature differences across the area sampled by the sensor.

Measurements are described which help to assess the precision and bias of the apparatus. These measurements involve the use of standard reference materials and participation in round-robin measurement programs. Data from this apparatus are compared with measurements on similar materials reported in the literature.

Error-propagation analysis suggests the estimated imprecision in measurement of thermal conductivity is 0.6%. The experimentally observed imprecision under dynamic control of the automated system is 1% near room temperature, rising to 5% at the upper end of the temperature range. The reproducibility of the apparatus is found experimentally to be about 1.2%. Participation in a round-robin measurement program on fibrous board insulations showed results from this apparatus to lie within 5% and 7% of the means of all participating laboratories for two different materials. No great weight should be given to this deviation since specimens measured by the round-robin participants were different from the ones we measured. Measurement by an industrial laboratory on one similar material agreed within 4% of the results from this apparatus.

7. REFERENCES

[1] Siu, M.C.I., "Fibrous Glass Board as a Standard Reference Material for Thermal Resistance Measurement Systems", Symposium on Thermal Insulation Performance, ASTM STP 718, pp. 343-360 (ASTM, 1980).

[2] Smith, D.R. and Hust, J.G., "Effective Thermal Conductivity of a Glass Fiber-Board Standard Reference Material", NBSIR 81-1639, (U.S. National Bureau of Standards, 1981).

[3] Siu, M.C.I. and Hust, J.G., "Standard Reference Material 1450b, Thermal Resistance-Fibrous Glass Board", National Bureau of Standards Certificate (Office of Standard Reference Materials, NBS, Gaithersburg, MD, 1982).

[4] Smith D.R. and Hust, J.G., "Effective Thermal Conductivity of Glass-Fiber Board and Blanket Standard Reference Materials", Thermal Conductivity 17: Proceedings of the 17th International Conference on Thermal Conductivity, Hust, J.G., Ed., pp. 408-410 (Plenum, NY, 1983).

[5] Hust, J.G., "Standard Reference Materials: Glass Fiberboard SRM for Thermal Resistance", NBS Special Publication 260-98 (U.S. National Bureau of Standards, Aug. 1985).

[6] Hust, J.G., "Standard Reference Materials: Glass Fiberblanket SRM for Thermal Resistance", NBS Special Publication 260-103 (U.S. National Bureau of Standards, Sep. 1985).

[7] Smith, D.R., Hust J.G. and Van Poolen, L.J., "A Guarded-hot-Plate Apparatus for Measuring Effective Thermal Conductivity of Insulations Between 80 K and 360 K", NBSIR 81-1657 (U.S. National Bureau of Standards, 1982).

[8] Standard Test Method C 177 for "Steady-State Heat Flux Measurements and Thermal Transmission Properties by Means of the Guarded-Hot-Plate Apparatus", 1986 ANNUAL BOOK OF ASTM STANDARDS, Vol. 04.06, pp. 21-36, (ASTM, Philadelphia, 1986).

[9] Standard Practice C 1045-85 for "Calculating Thermal Transmission Properties from Steady-State Heat Flux Measurements", 1986 ANNUAL BOOK OF ASTM STANDARDS, Vol. 4.06, pp. 689-696, (ASTM, Philadelphia, 1986).

[10] Standard Practice C 1044-85 for "Using the Guarded-Hot-Plate Apparatus in the One-Sided Mode to Measure Steady-State Heat Flux and Thermal Transmission Properties", 1986 ANNUAL BOOK OF ASTM STANDARDS, Vol. 4.06, pp. 685-688 (ASTM, Philadelphia, 1986).

[11] Hust, J.G. and Lankford, A.B., "Comments on the Measurement of Thermal Conductivity and Presentation of a Thermal Conductivity Integral Method", Int. J. of Thermophysics 3/1, 67-77 (1982).

[12] Hust, J.G. and Smith, D.R., "Round-Robin Measurements of the Apparent Thermal Conductivity of Two Refractory Insulation Materials Using High-Temperature Guarded-Hot-Plate Apparatus" NBSIR 88-3087, (U.S. National Bureau of Standards, May 1988)

[13] Mitchell, H., "The Development of a Refractory Fiber Insulation for Use As a High Temperature Thermal Transmission Calibration Sample". In press: Proceedings of the 19th International Conference on Thermal Conductivity, Yarbrough, D., Ed. (Plenum, N.Y., 1988).

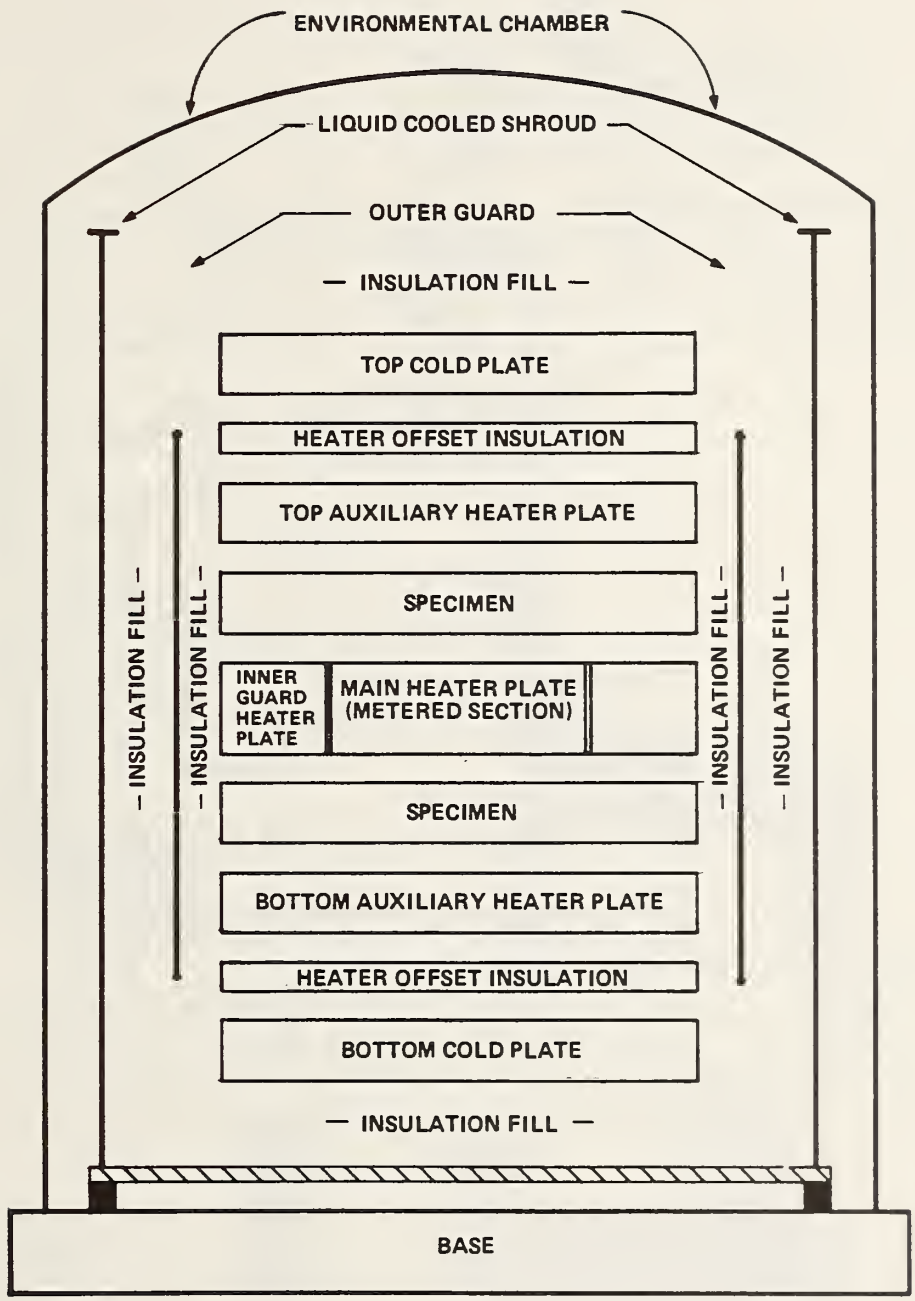

Figure 1. Layout of thermal conductivity stack, guards, shroud and environmental chamber of the NBS high-temperature guarded-hot-plate apparatus

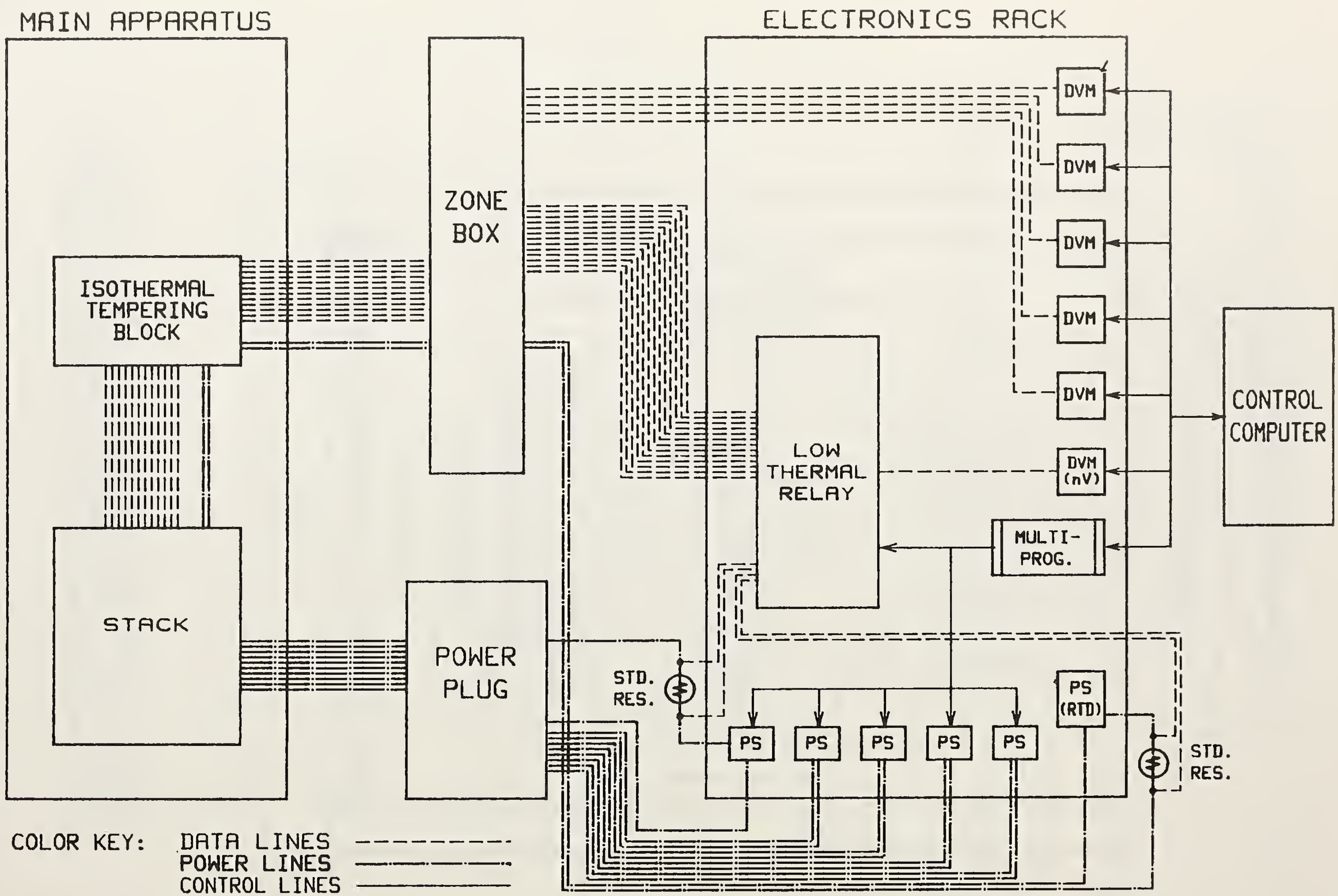

Figure 2. Block diagram of electronic system for control of temperature and acquisition of data

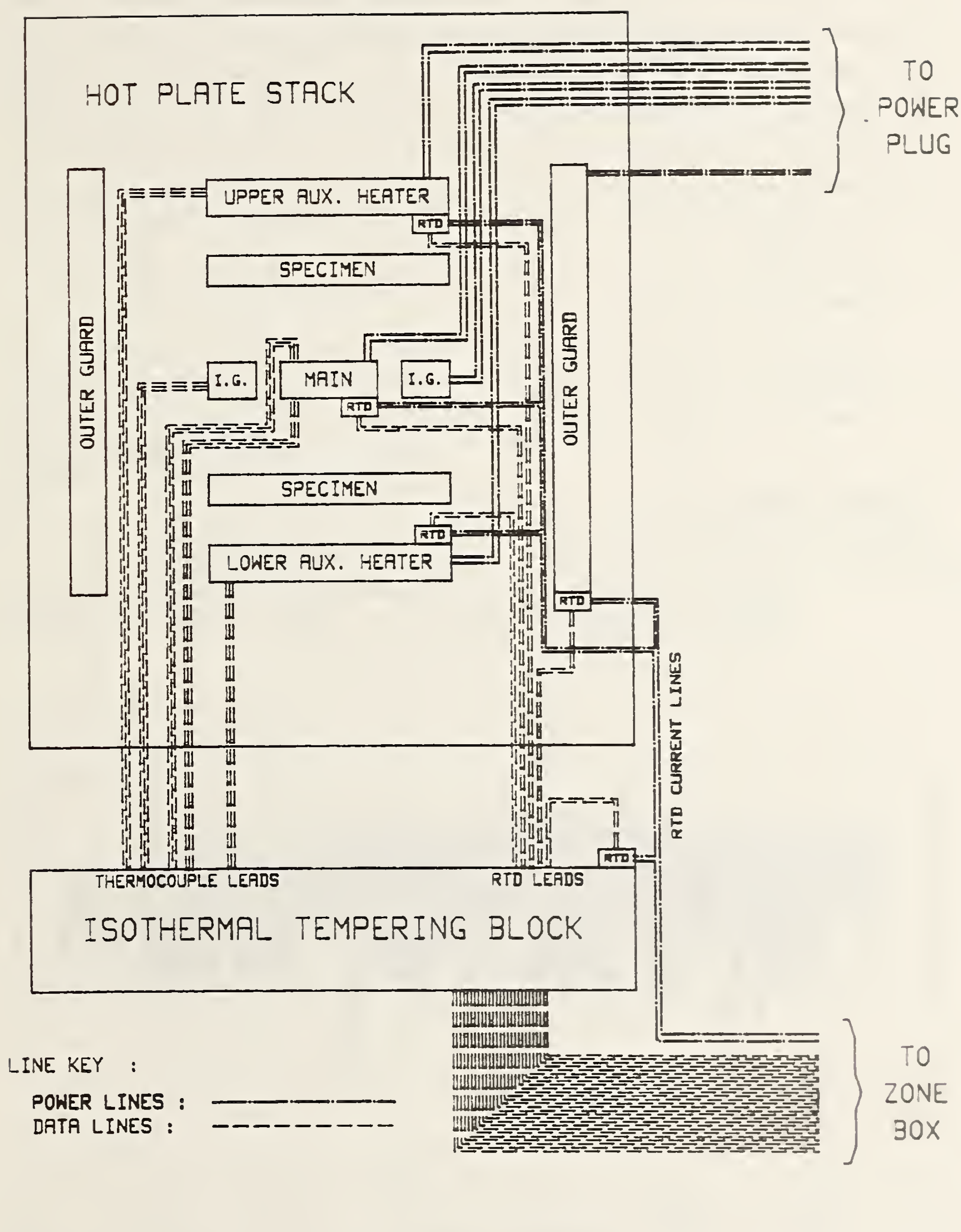

Figure 3. Detailed block diagram of control and data acquisition within the "main apparatus" block of Figure 2

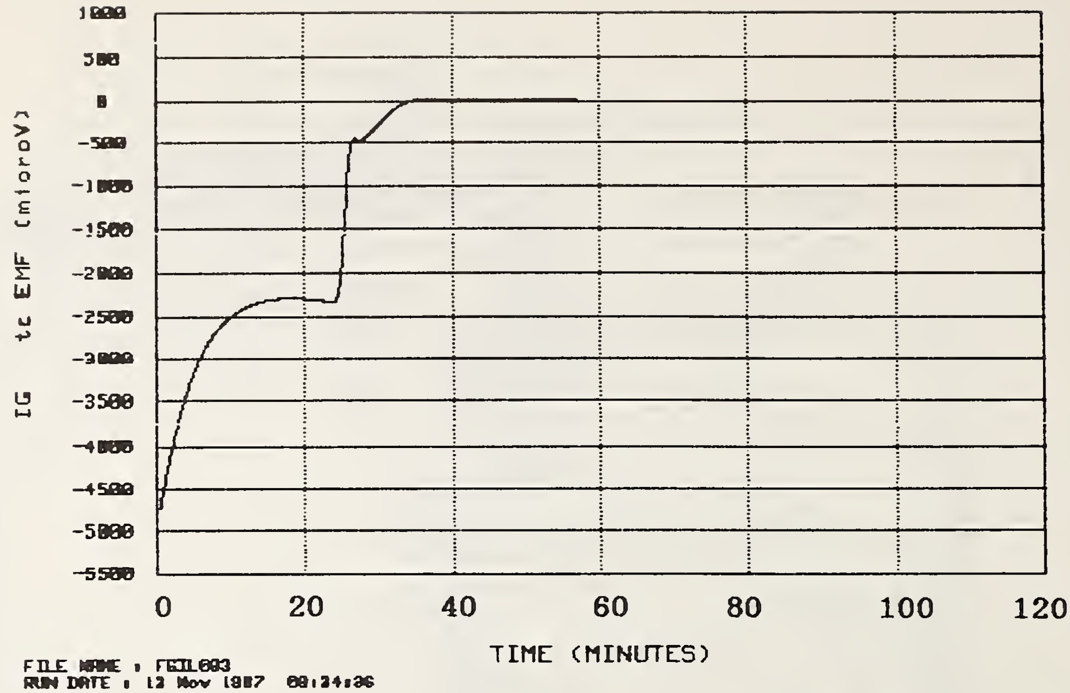

Figure 4. Output emf of gap thermocouple, between inner guard and main heater, vs. time: behavior during initial approach to set-point

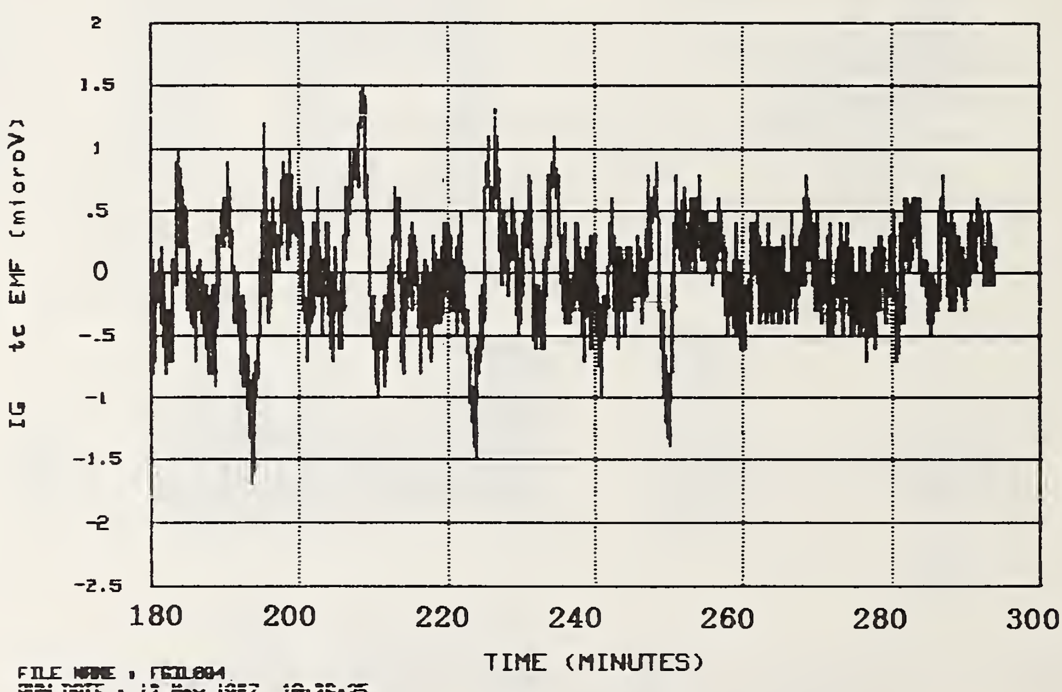

Figure 5. Output emf of gap thermocouple, between inner guard and main heater, vs. time: behavior during stable operation at set-point

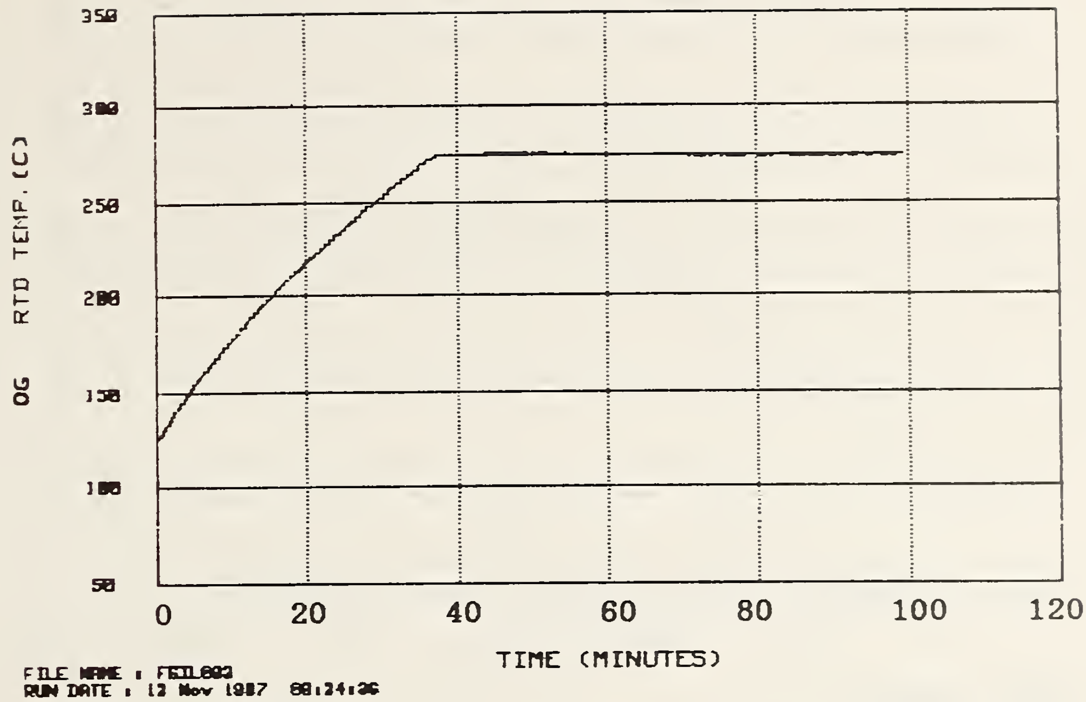

Figure 6. Temperature of outer guard, measured by resistance thermometer, vs. time: behavior during initial approach to set-point

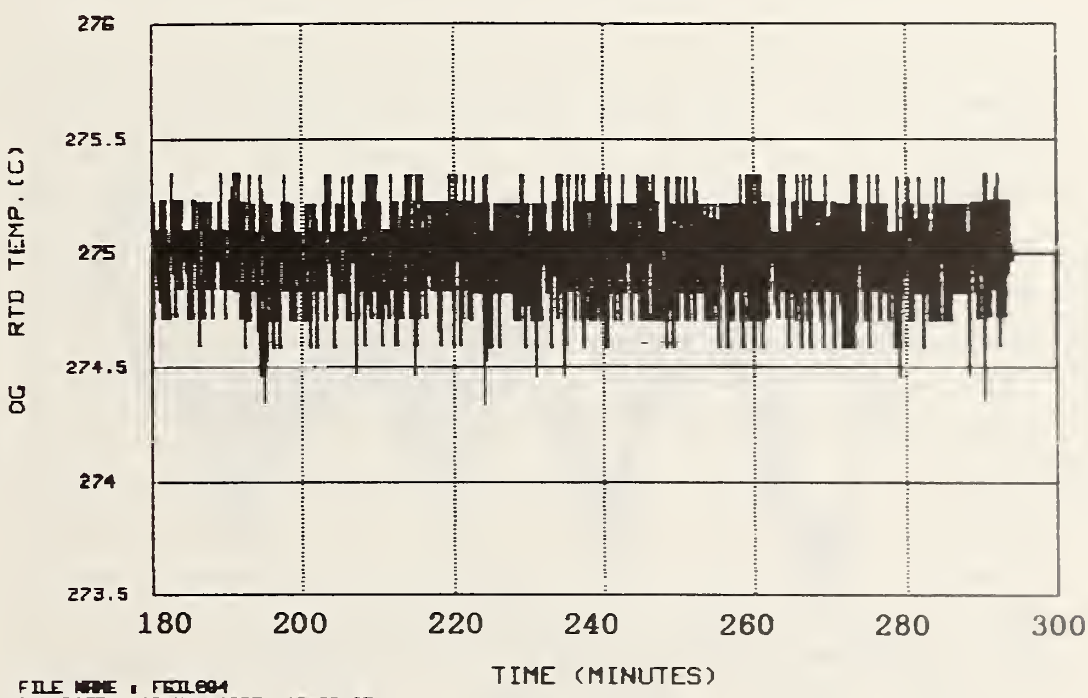

Figure 7. Temperature of outer guard, measured by resistance thermometer, vs. time: behavior during stable operation at set-point

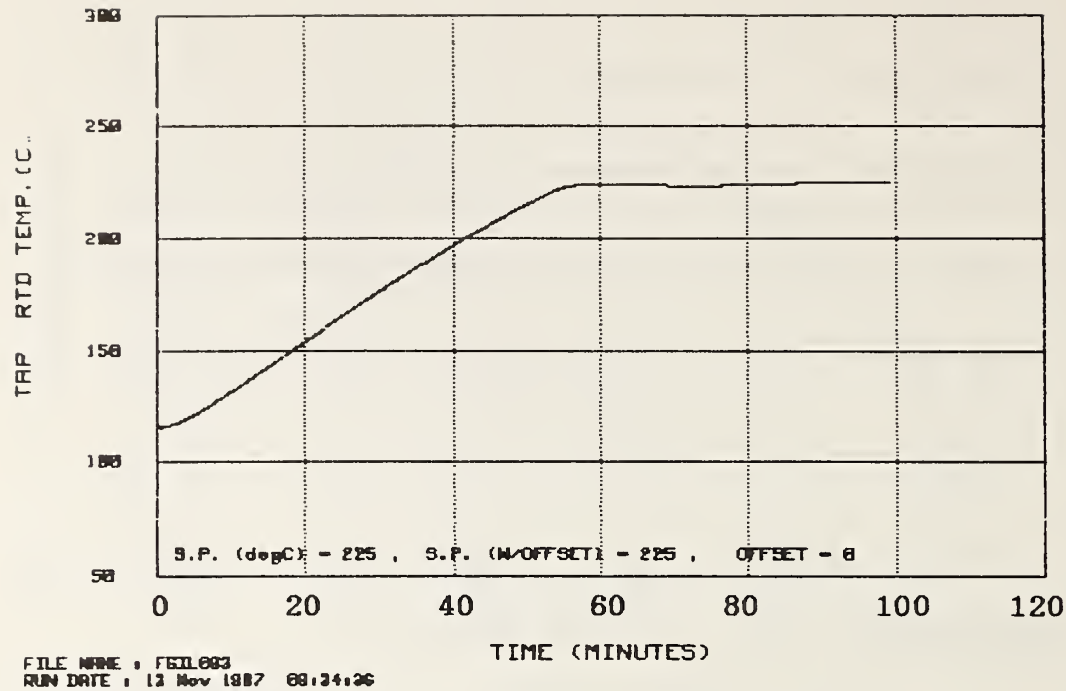

Figure 8. Temperature of top auxiliary heater plate, measured by resistance thermometer, vs. time: behavior during initial approach to set-point

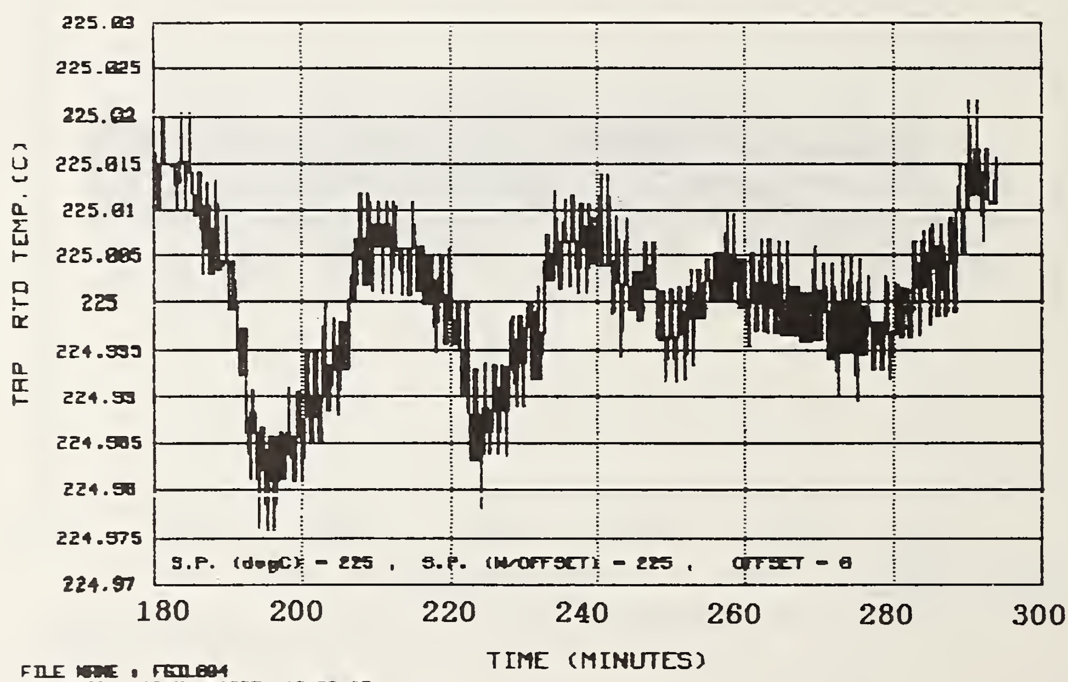

Figure 9. Temperature of top auxiliary heater plate, measured by resistance thermometer, vs. time: behavior during stable operation at set-point

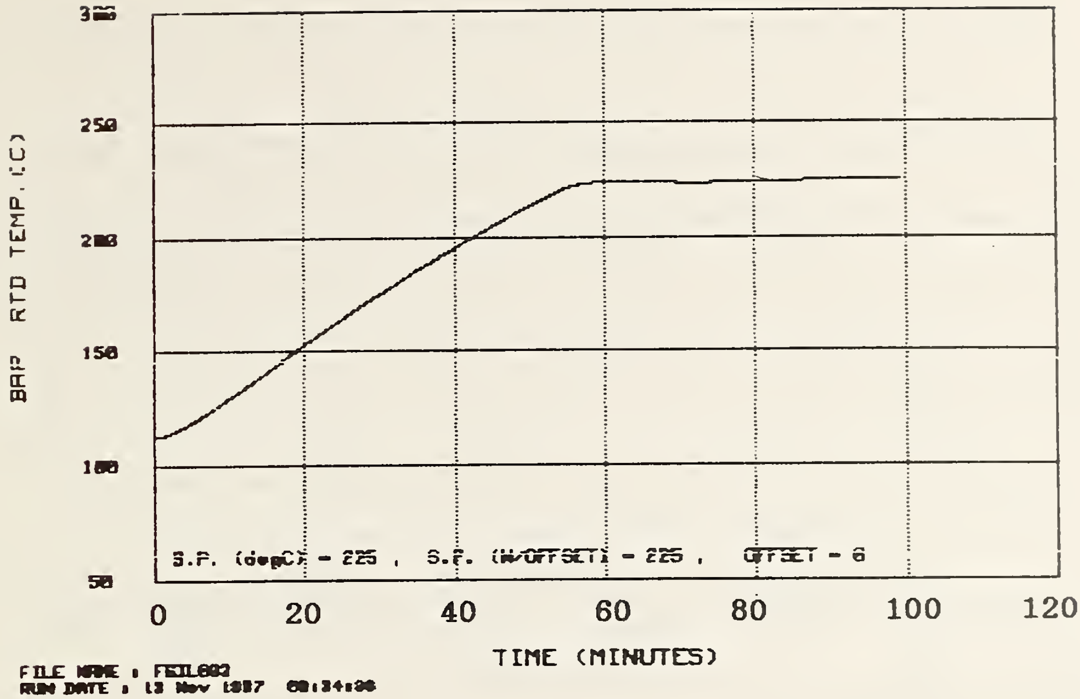

Figure 10. Temperature of bottom auxiliary heater plate, measured by resistance thermometer, vs. time: behavior during initial approach to set-point

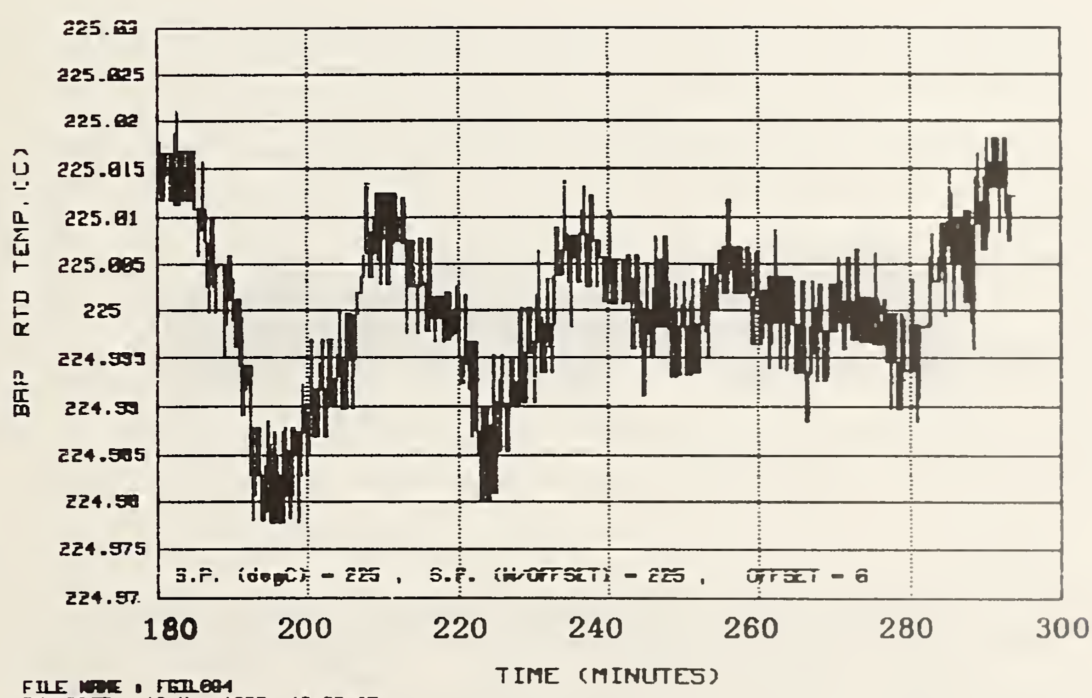

Figure 11. Temperature of bottom auxiliary heater plate, measured by resistance thermometer, vs. time: behavior during stable operation at set-point

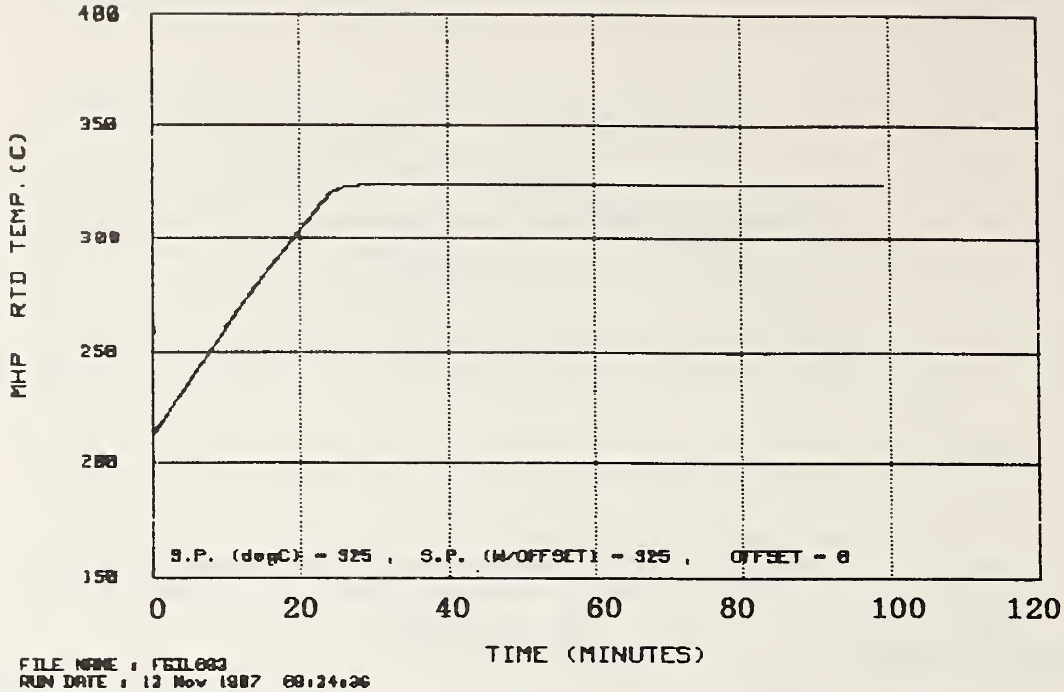

Figure 12. Temperature of main heater plate, measured by resistance thermometer, vs. time: behavior during initial approach to set-point

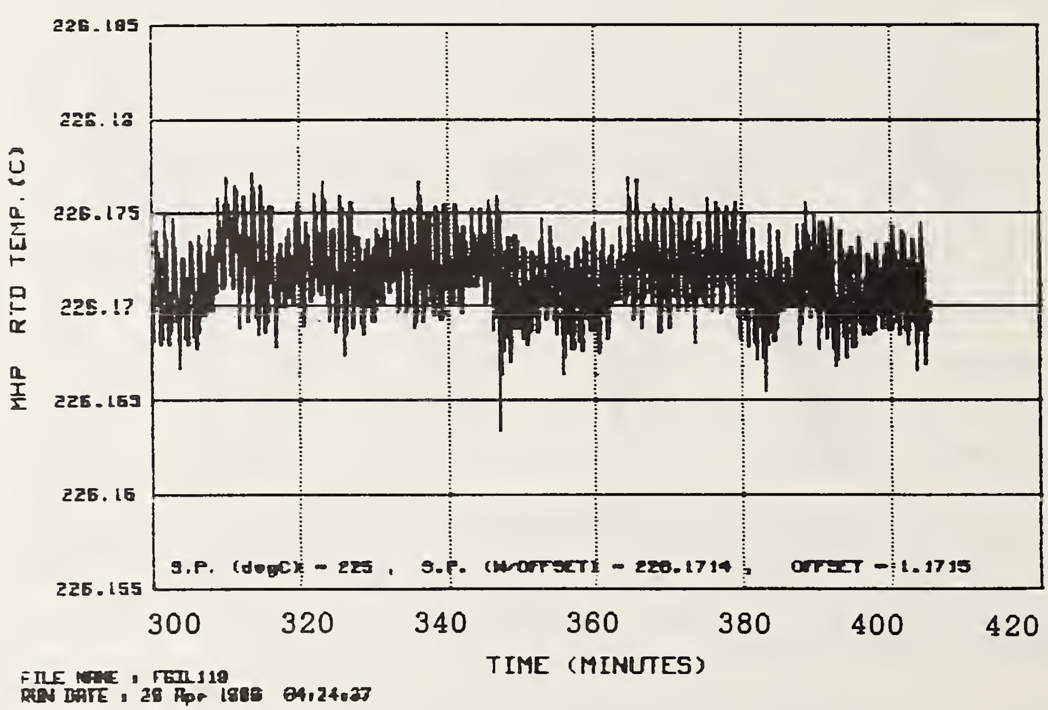

Figure 13. Temperature of main heater plate, measured by resistance thermometer, vs. time: behavior during stable operation at set-point

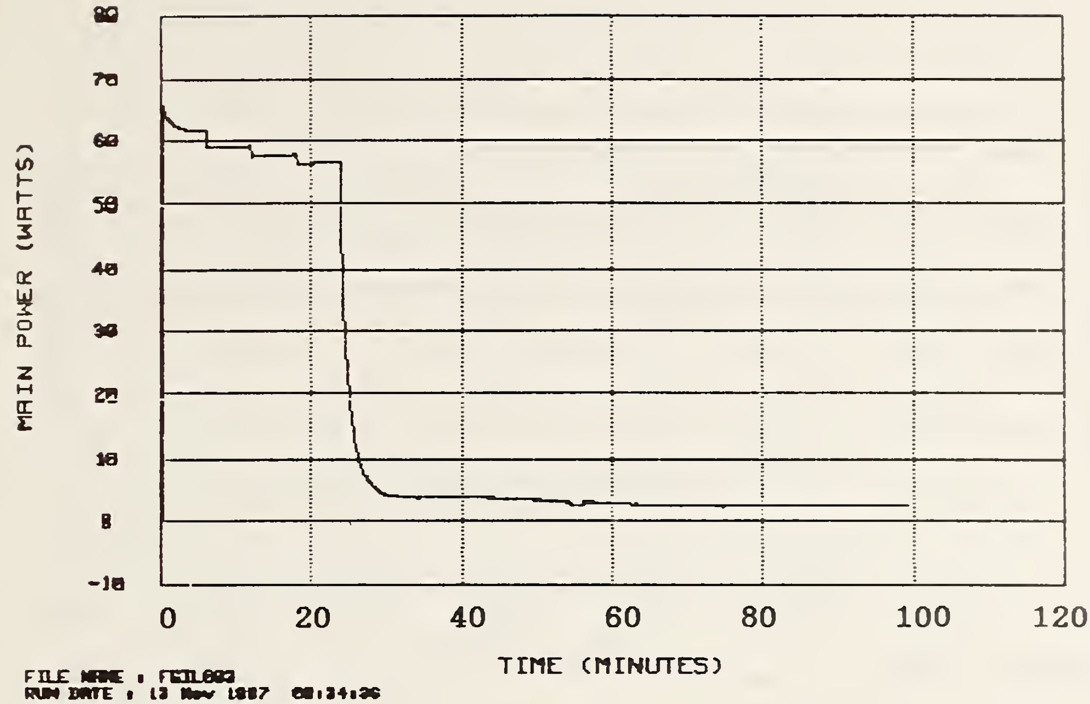

Figure 14. Main heater power vs. time: behavior during initial approach to set-point

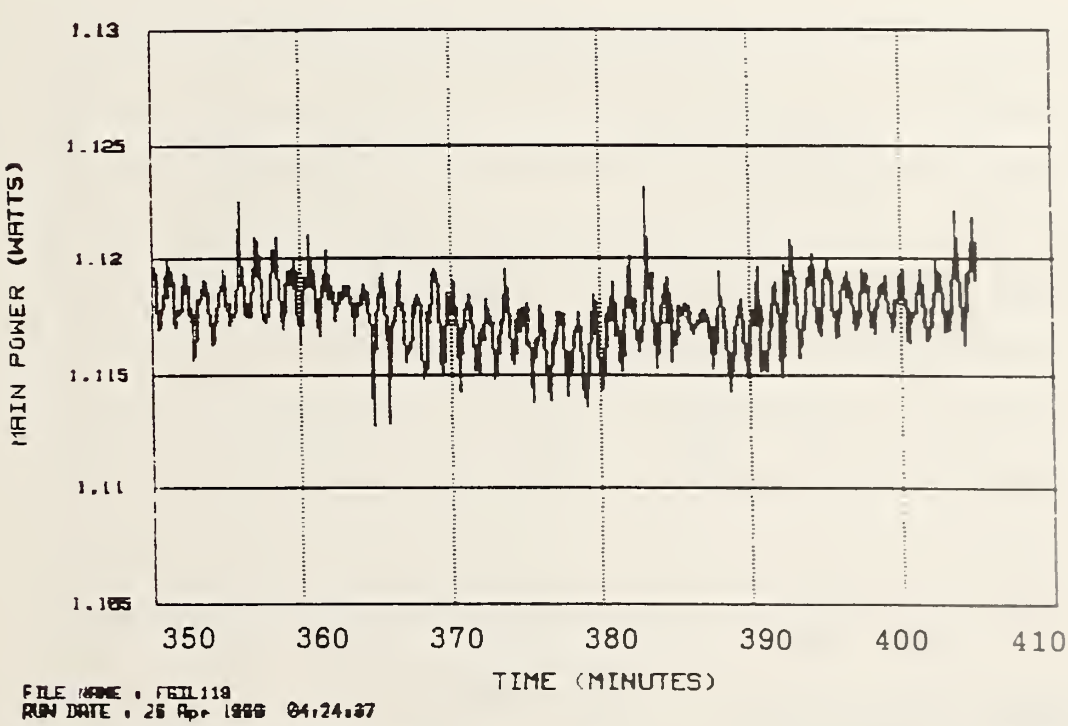

Figure 15. Main heater power vs. time: behavior during stable operation at set-point

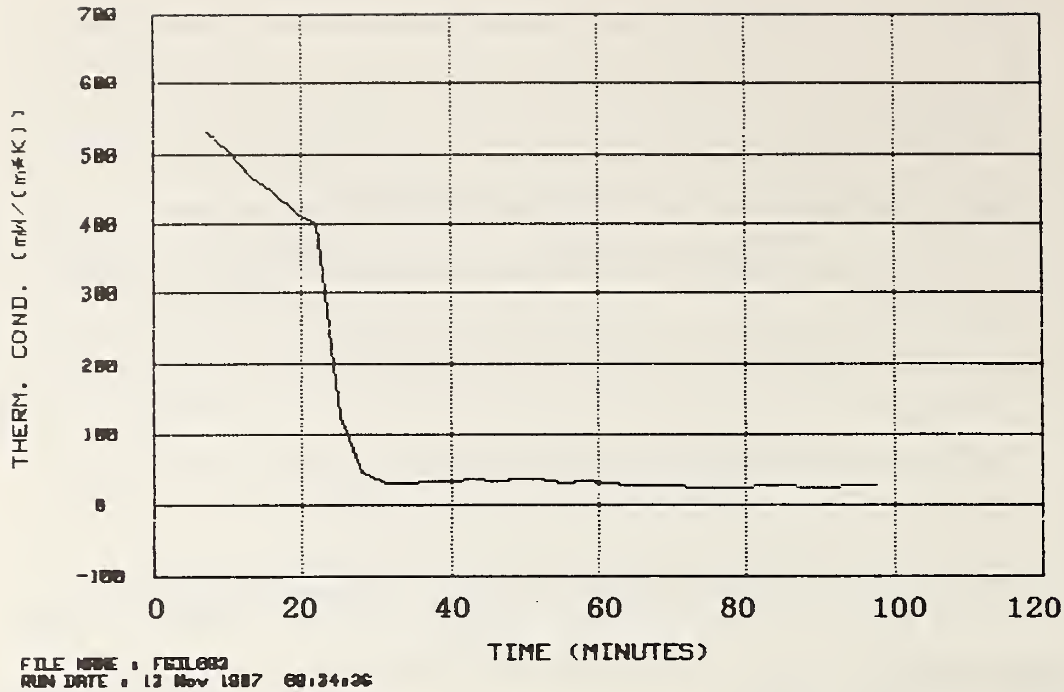

Figure 16. Experimental thermal conductivity, calculated from main heater power, specimen area and temperature gradient, vs. time: behavior during initial approach to set-point

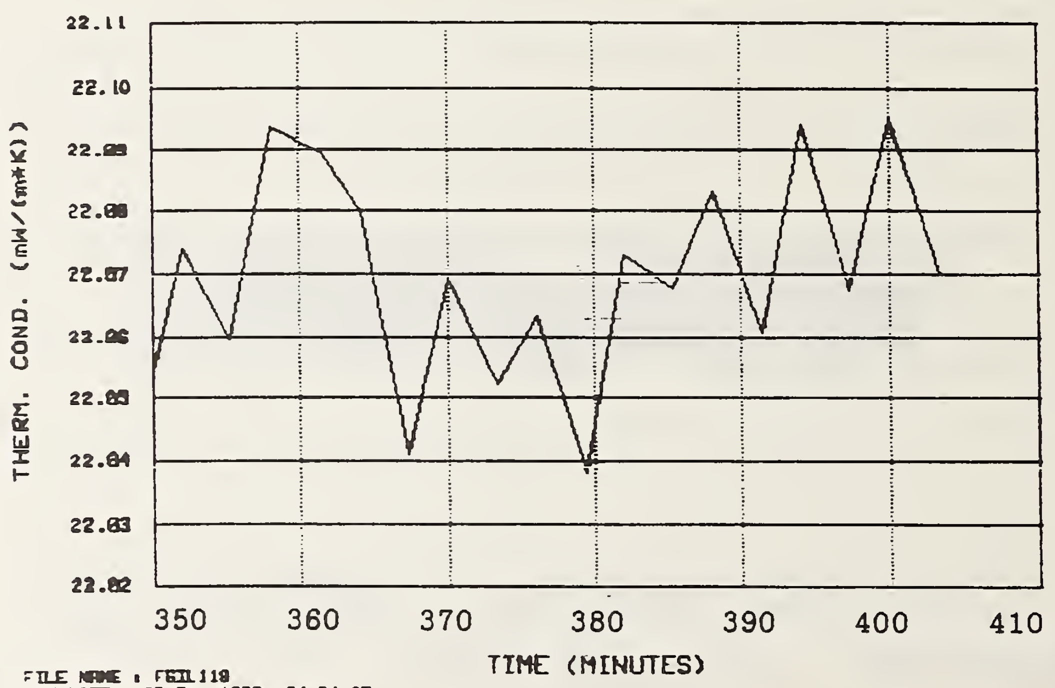

Figure 17. Experimental thermal conductivity, calculated from main heater power, specimen area and temperature gradient, vs. time: behavior during stable operation at set-point

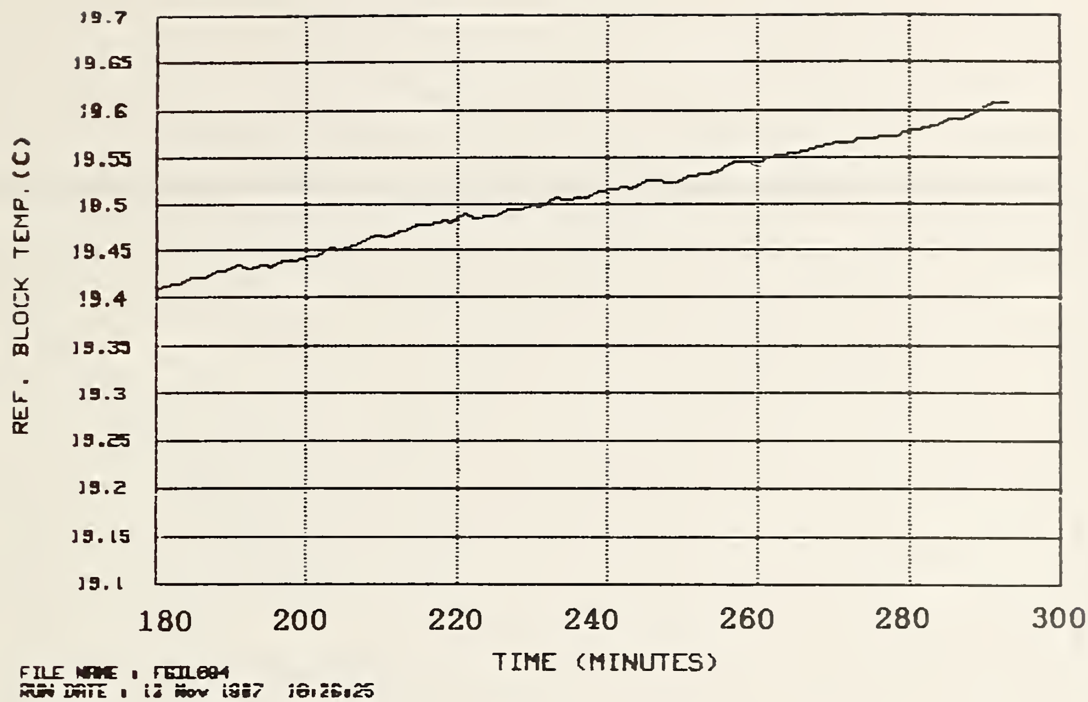

Figure 18. Temperature of thermocouple reference block, measured by resistance thermometer, vs. time

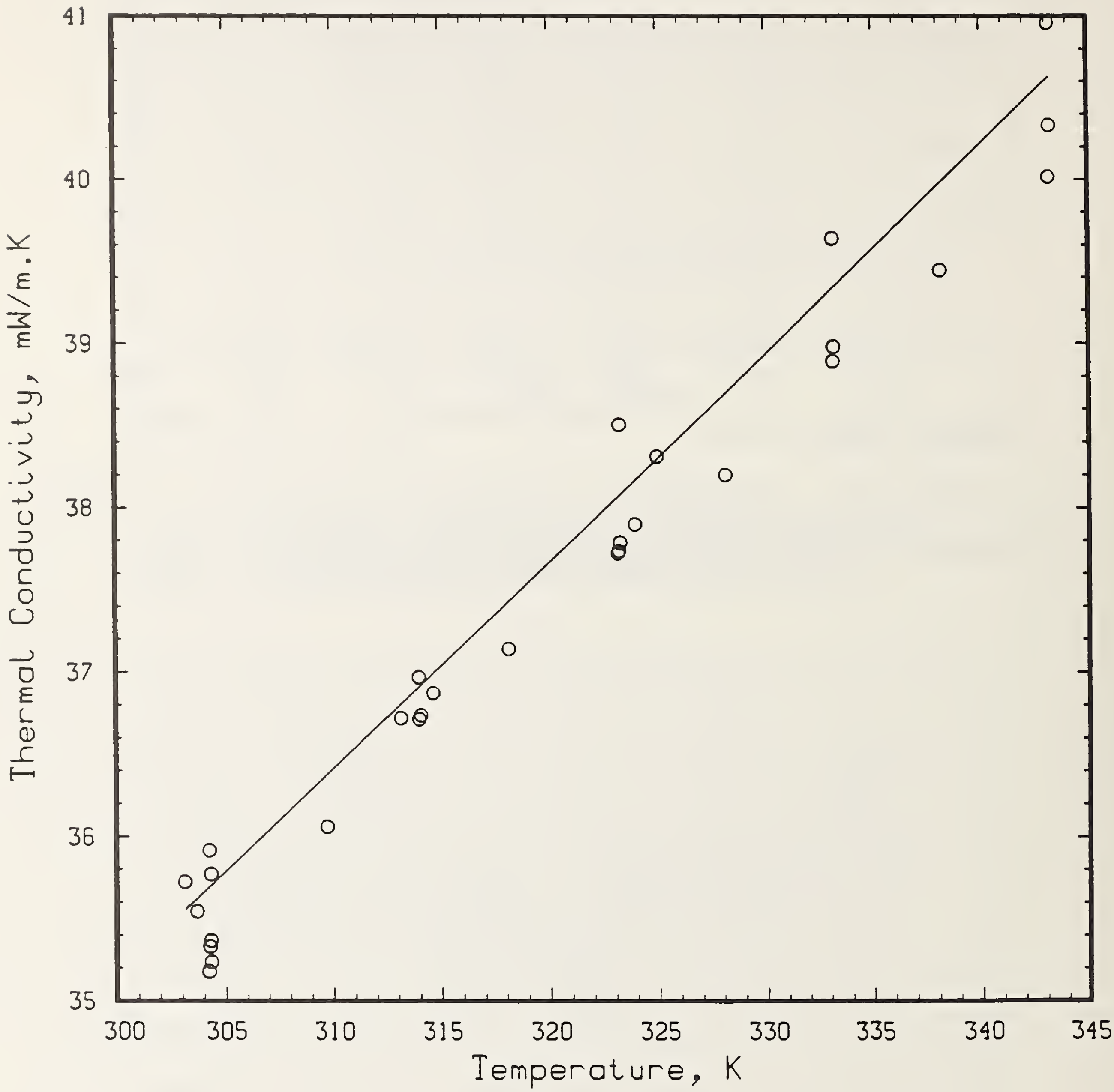

Figure 19. Thermal conductivity of fibrous glass insulation SRM 1450b compared with certification function for k(T) (solid line)

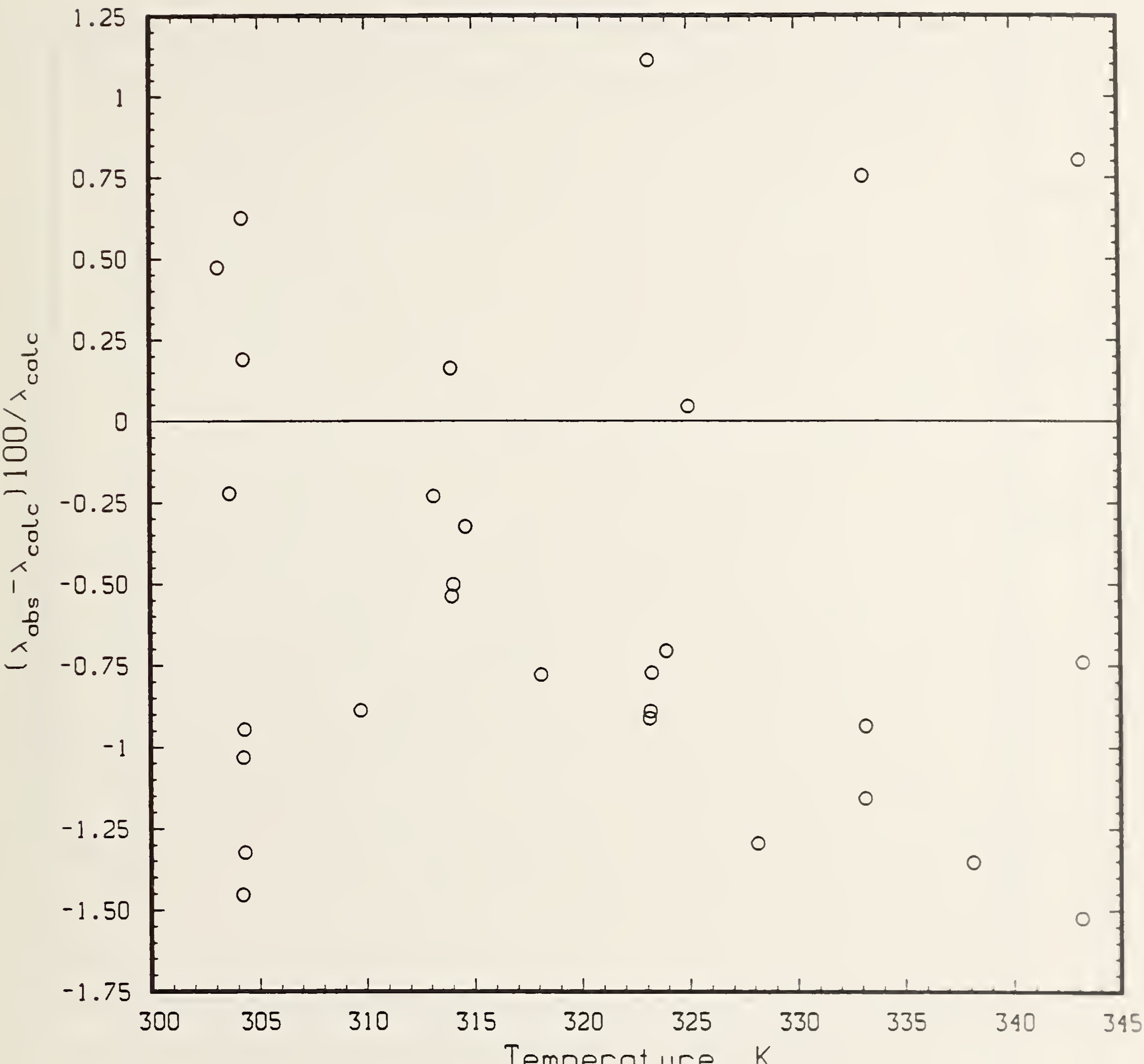

Figure 20. Relative deviations of thermal conductivity of fibrous glass insulation SRM 1450b, compared with certification function for k(T)

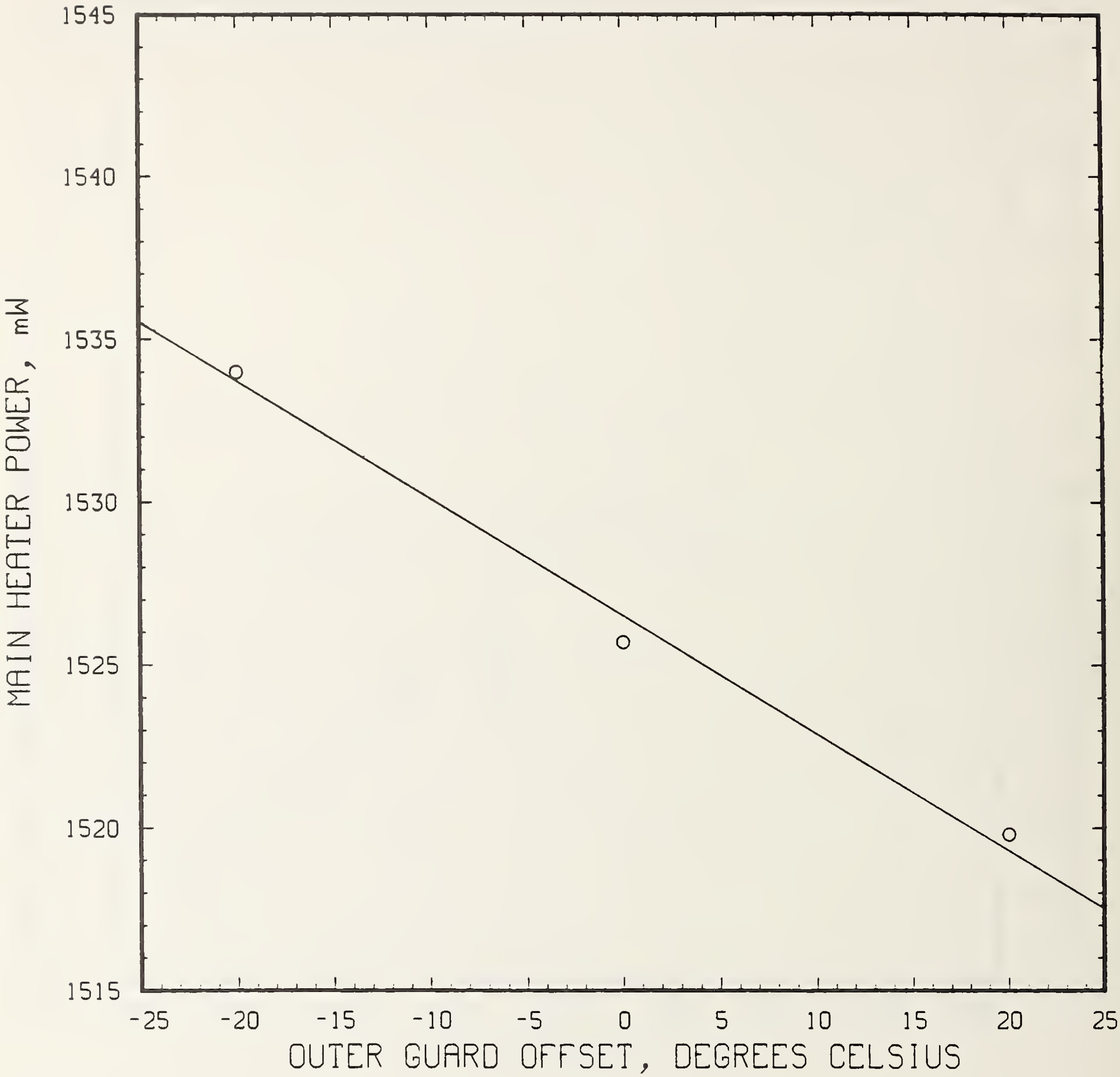

Figure 21. Main heater power supplied to specimens of fibrous alumina-silica insulation board, for outer-guard offsets of ± 20 K

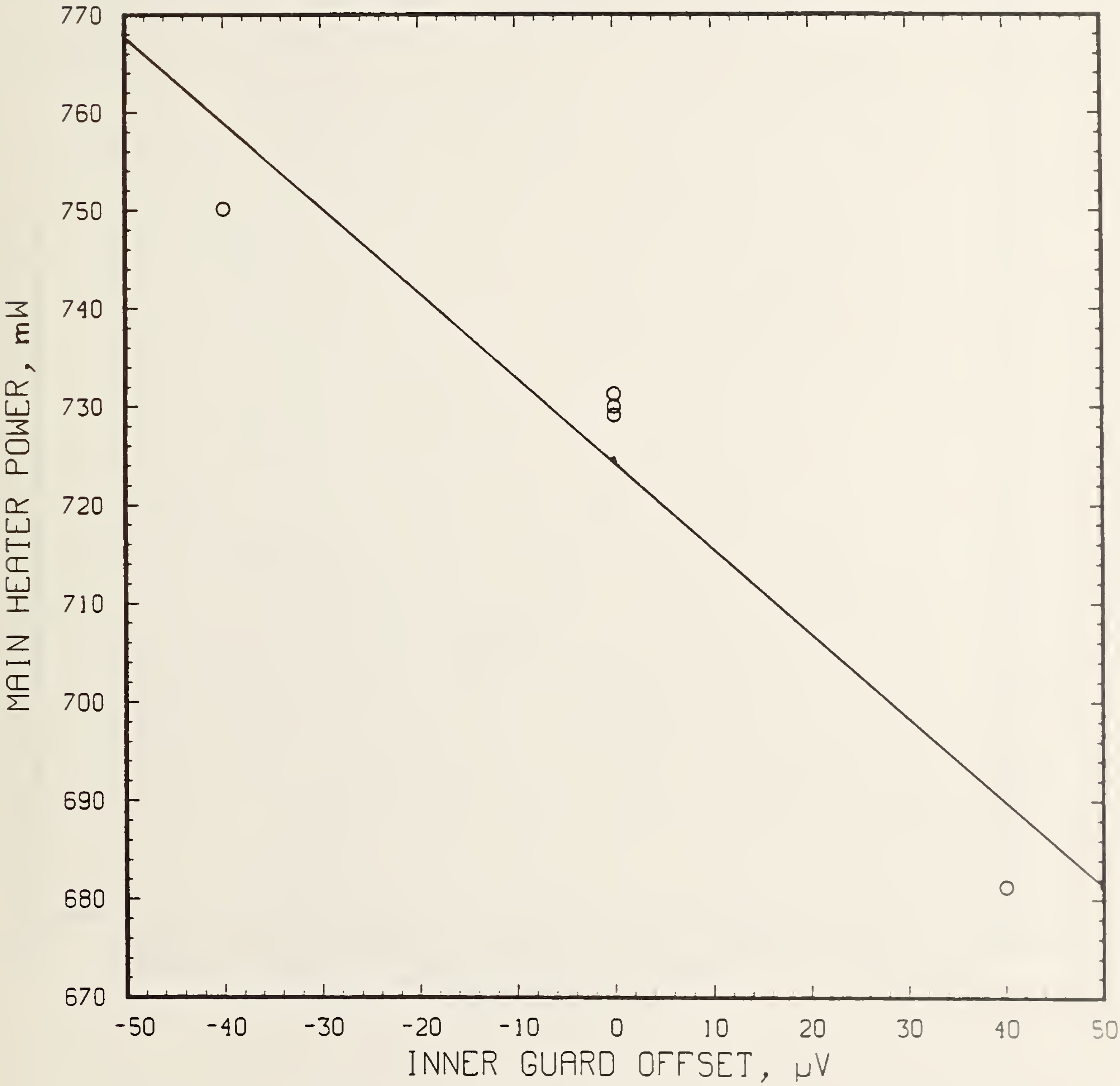

Figure 22. Main heater power supplied to specimens of fibrous glass insulation board for inner-guard offsets of ± 40 uV

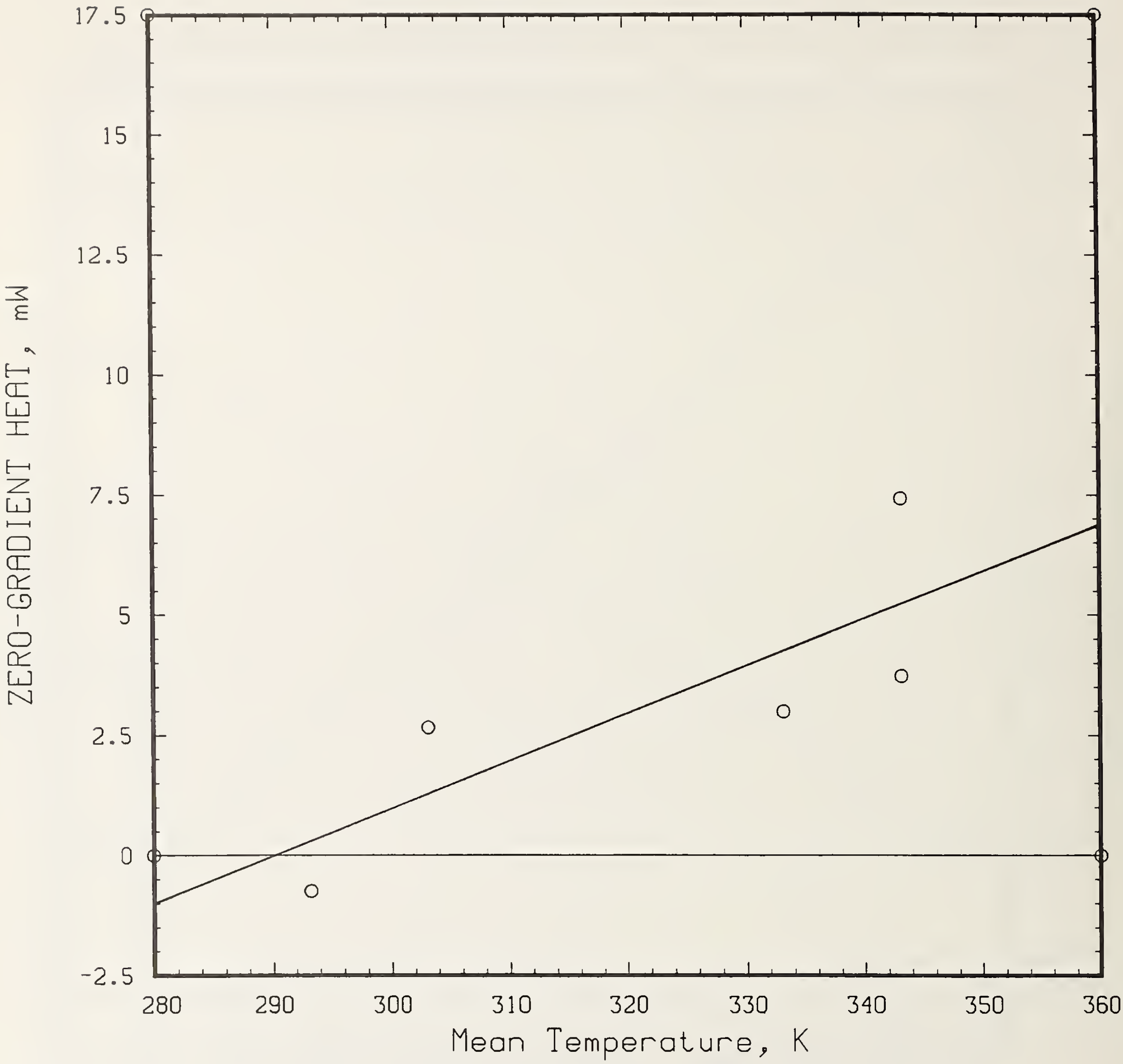

Figure 23. Zero-gradient heat vs. temperature, for fibrous glass insulation board

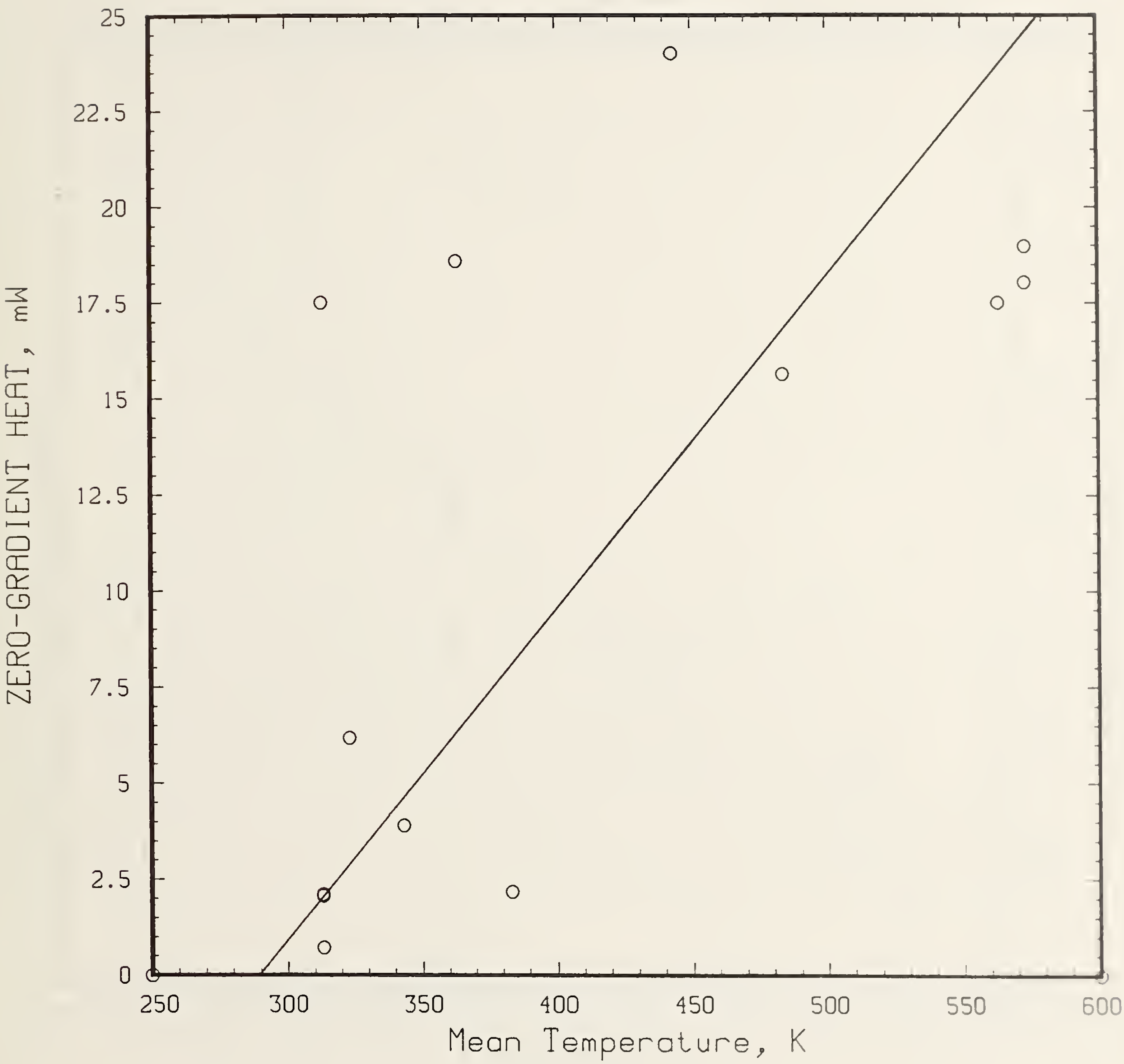

Figure 24. Zero-gradient heat vs. temperature, for fibrous alumina-silica insulation board

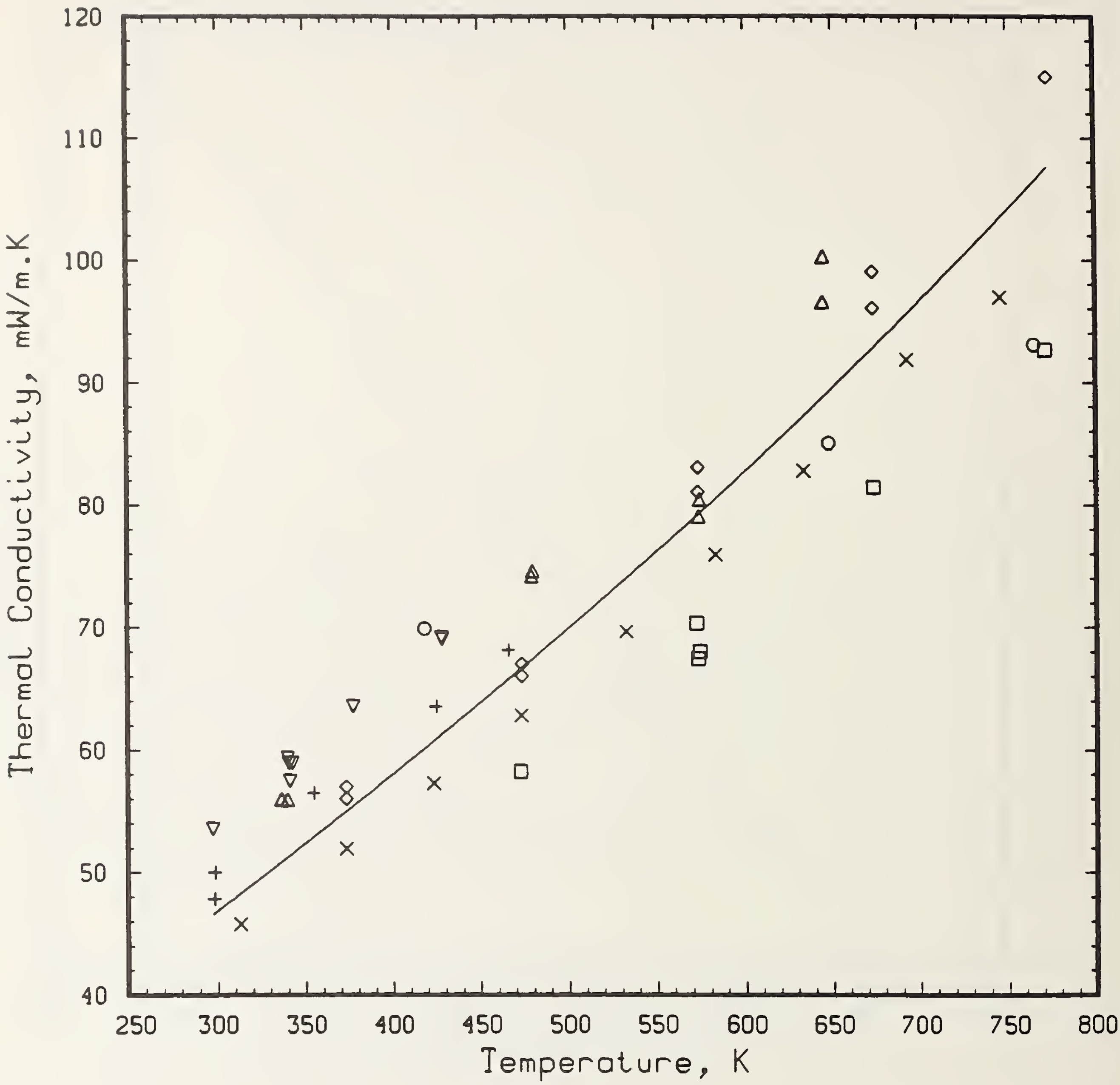

Figure 25. Thermal conductivity round-robin test results for fibrous alumina-silica. The solid curve is calculated from $k(T) = 15.98+0.1003T + 3.053x10^{-8}\ T^3$

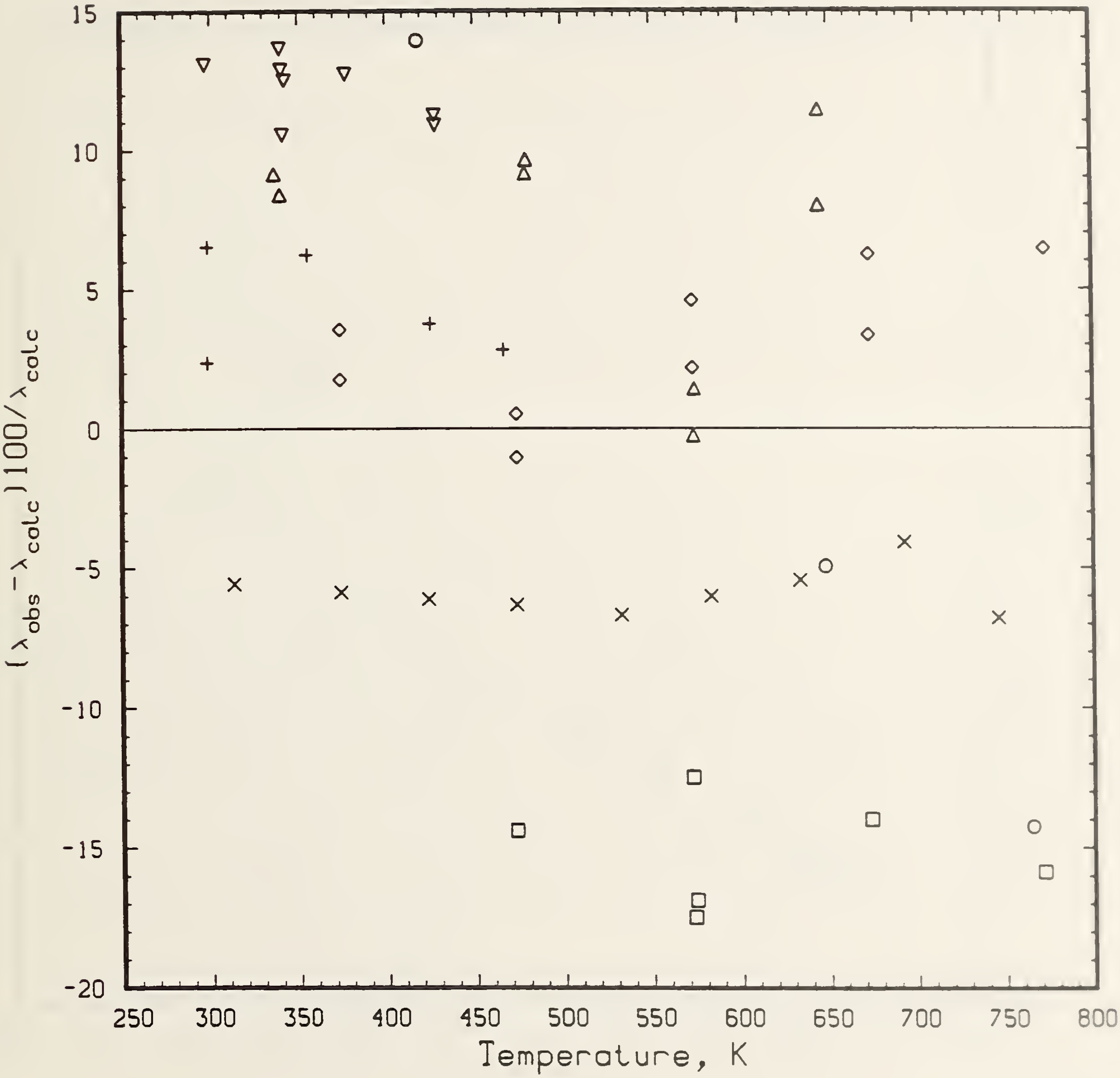

Figure 26. Deviations of thermal conductivity round-robin test results from values calculated for fibrous alumina-silica, using the relation given in caption to Figure 25

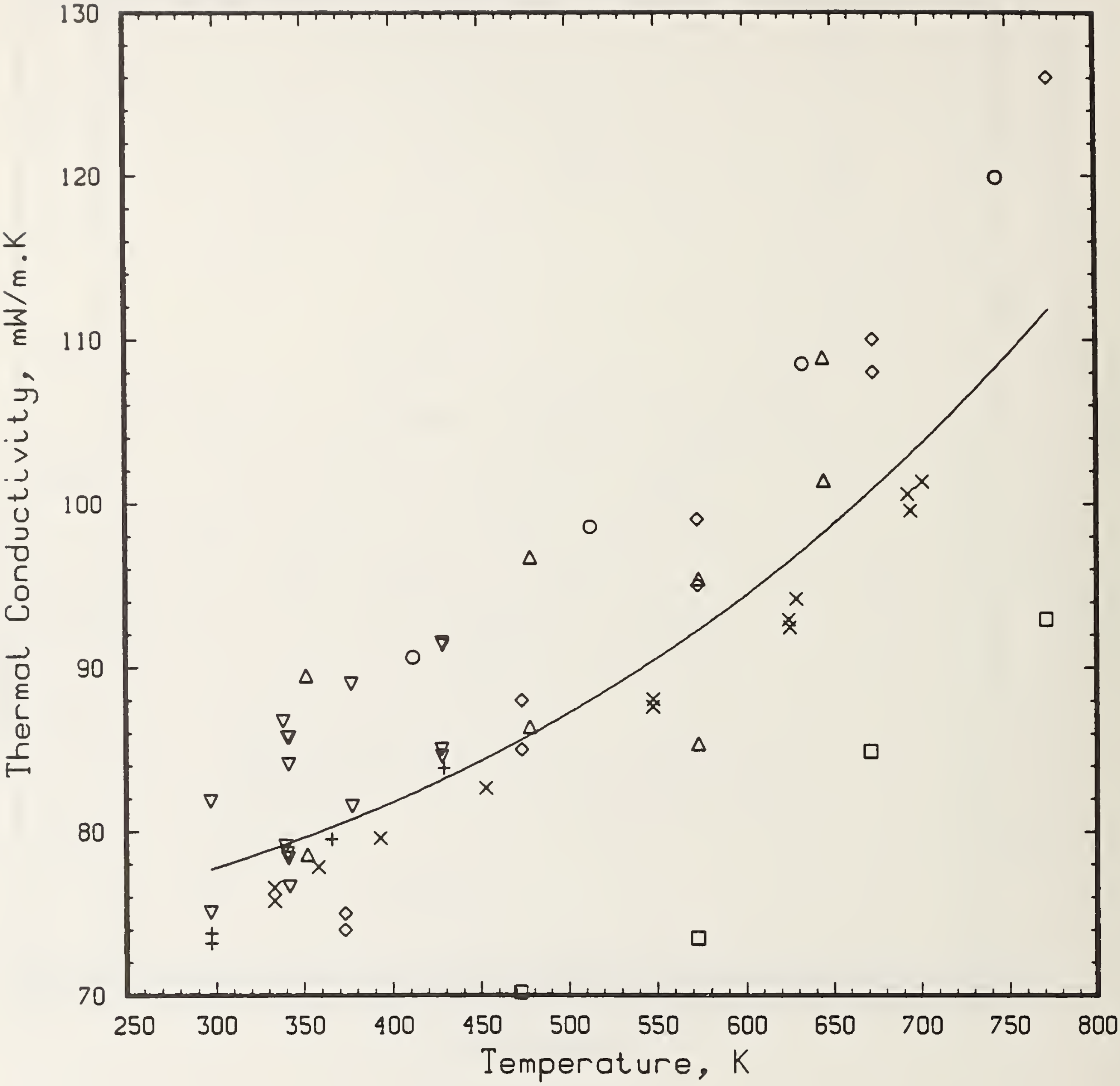

Figure 27. Thermal conductivity round-robin test results for calcium silicate. The solid curve is calculated from $k(T) = 70.67 + 0.01878\ T + 5.796\times10^{-8}\ T^3$

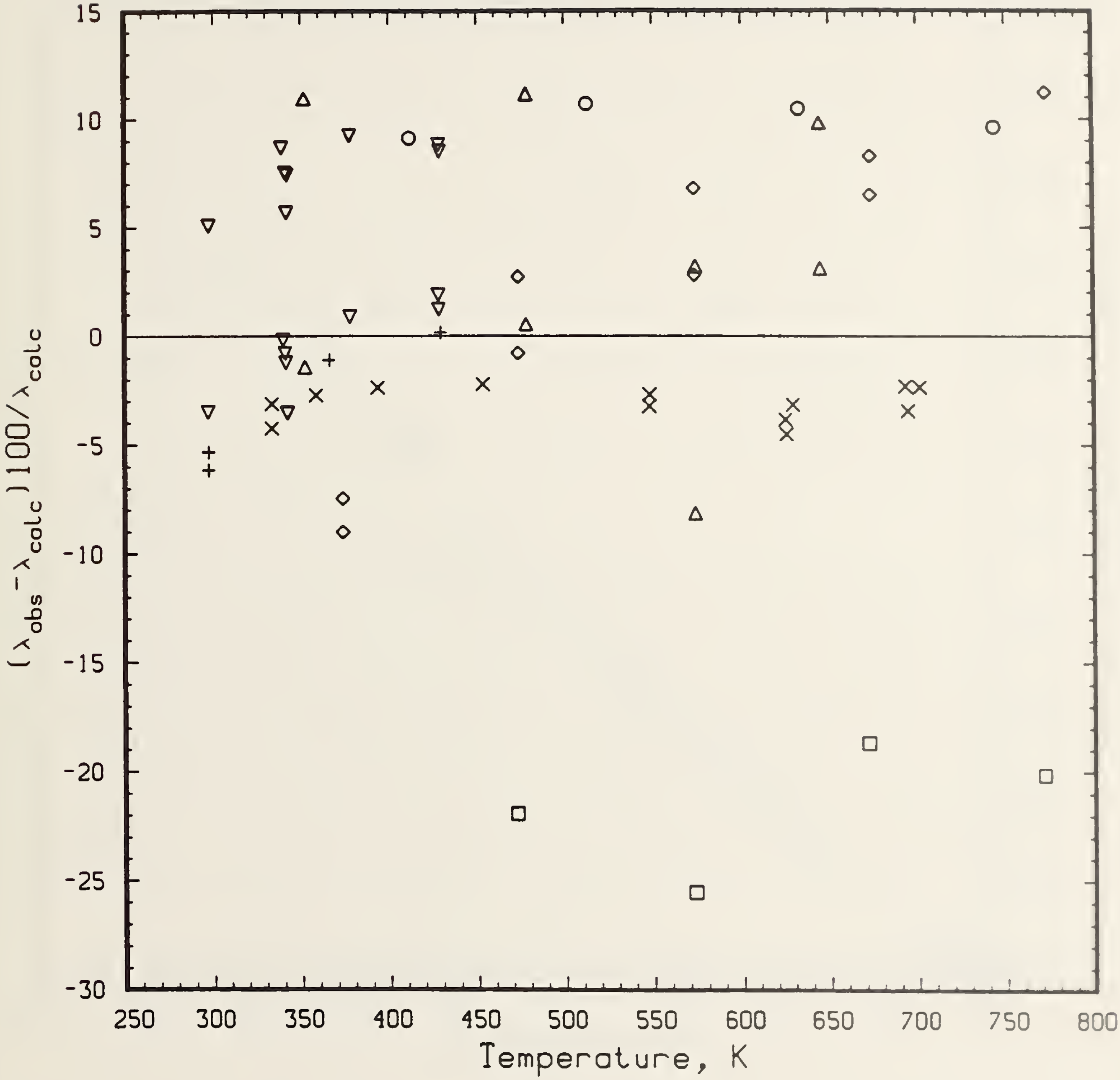

Figure 28. Deviations of thermal conductivity round-robin test results from values calculated for calcium silicate, using the relation given in caption to Figure 27

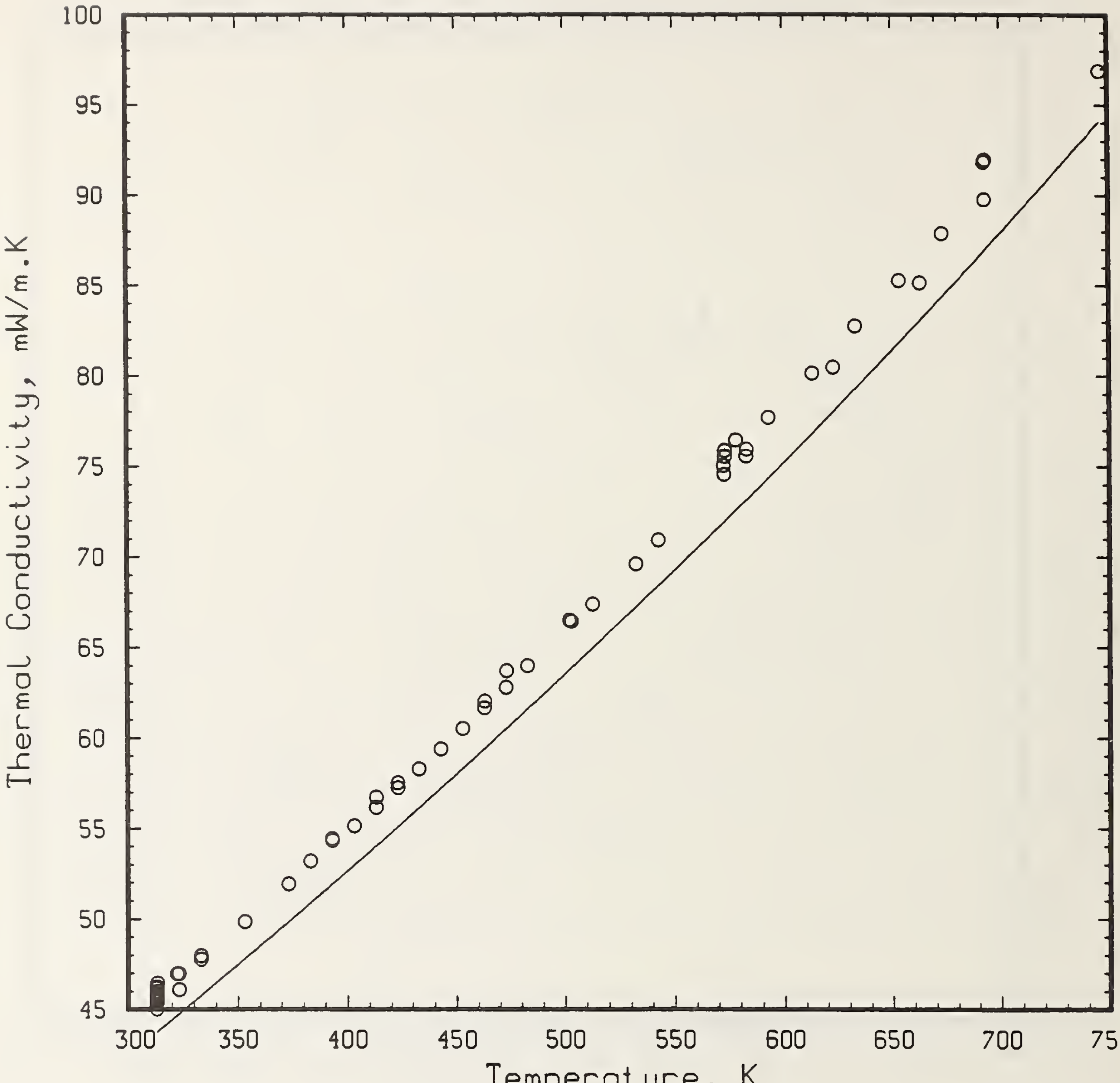

Figure 29. Thermal conductivity of refractory fibrous alumina-silica insulation board (circles) compared to functional correlation of Mitchell [8]

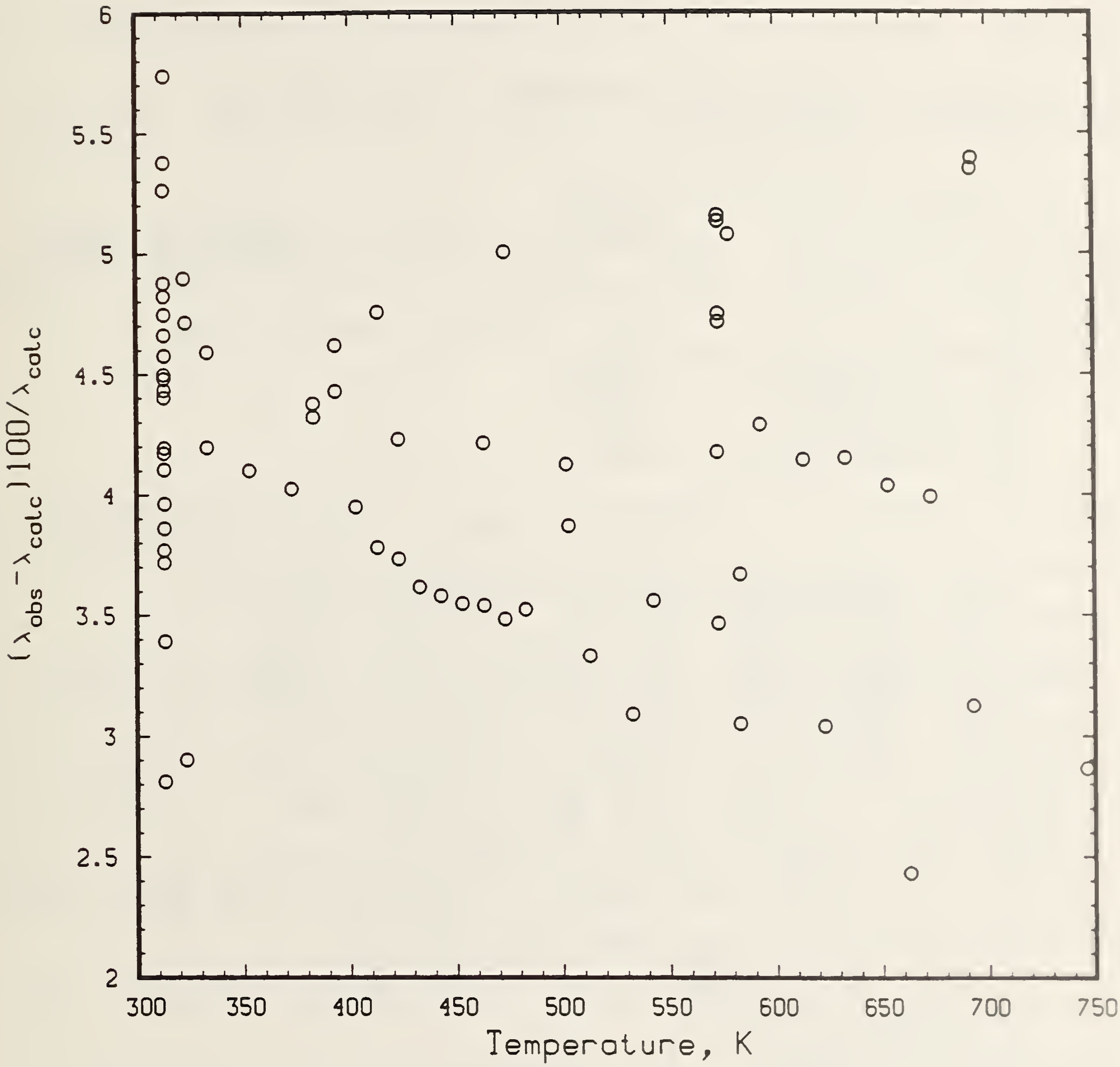

Figure 30. Relative deviation of thermal conductivity data from functional correlation of Mitchell

Table 1. Typical data summary as printed at end of a measurement sequence

```
FILE NAME : FSIL094

13 Nov 1987  10:26:25    OPERATING MODE : DOUBLE SIDED
 USER DETERMINED STABILITY AT : 1.3 min. ;  TOTAL RUN TIME : 293 min.
SRM CANDIDATE:  FUMED SILICA

        GENERAL FILE SPECIFICATIONS:
 SAMPLE THICKNESS (cm)    ; UNCORRECTED= 2.523 ,   CORRECTED= 2.533
 MAIN PLATE AREA (cm^2) ; UNCORRECTED= 123.505 ,   CORRECTED= 128.909
 AREA DENSITY (kg/m^2)= 7.704 ,  CORRECTED BULK DENSITY (kg/m^3)= 304.064
 PLATE SPACER MATERIAL : STAINLESS STEEL
 FILL GAS : AIR                        ,    PRESSURE (mmHg) =  629
 SPECIMEN AND SPECIMEN GUARD CODES :
       TOP SPECIMEN CODE    : FUSIL-1
       TOP GUARD CODE       : NONE
       BOTTOM SPECIMEN CODE : FUSIL-2
       BOTTOM GUARD CODE    : NONE

 CORRECTED AVERAGE PLATE TEMPERATURES (degC)  AND TEMP. CORRECTIONS:

    UPPER AUX. PLATE TEMP.= 226.983 , STD.DEV.= .007
        TEMP. CORRECTION=-.0971 ,  dT/dt(deg/hr)= .0326
    UPPER MAIN PLATE TEMP.= 324.5048 , STD.DEV.= .0056
        TEMP. CORRECTION=-.1753 ,  dT/dt(deg/hr)= .0318
    LOWER MAIN PLATE TEMP.= 324.3238 , STD.DEV.= .0058
        TEMP. CORRECTION=-.0876 ,  dT/dt(deg/hr)= .0339
    LOWER AUX. PLATE TEMP.= 226.587 , STD.DEV.= .0045
        TEMP. CORRECTION=-1.2749 ,  dT/dt(deg/hr)= .0016
    UPPER delta T= 97.5217 , STD. DEV. OF MEAN= .0027
    LOWER delta T= 97.7368 , STD. DEV. OF MEAN= .0022
    TOTAL delta T (T2-T1+T3-T4)= 195.2586
       STD. DEV. OF DelT = .0117 ,     % STD.DEV. OF DelT = 0
       STD. DEV. OF THE MEAN= .0035 ,    % STD.DEV. OF THE MEAN = .0018

 AVG. MAIN HEATER PLATE POWER (mW)= 2398.518
        STD.DEV. OF Q = 13.354 ,   % STD.DEV. OF Q = .55
        STD.DEV. OF THE MEAN = .702 ,    % STD.DEV. OF THE MEAN = .029
        dP/dt(mW/hr)=-6.977
 DATA AVERAGING INTERVAL :  30  MINUTES
 START POINT OF INTERVAL :  262.1  MINUTES

 FINAL HEATER RESISTANCE :  5.1361  OHMS

 IG THERMOPILE FINAL READINGS :
          UPPER PILE :  11.5  microvolts
          LOWER PILE : -9  microvolts
          TOTAL PILE :  2.1  microvolts

 Thi (degC) =  324.4143  ,  Tlo (degC) =  226.785  ,  delT= 97.6293
 AVG. TEMP.(C) IS :  275.600 ,  THERMAL COND. (k) =    24.14 mW/(m*K
        STD.DEV. OF k = .134 ,   % STD.DEV. OF k = .55
```

Table 2. Estimates of random variations and systematic uncertainties in measured quantities at room temperature.

A. Measurements at 325 K (52°C)

(R_H = 3.73 ohm; TC sensitivity: 29 μV/K)

Primary Variable	Value	Uncertainty	Imprecision	Bias
V	2.38 V	10 μV	10^{-3} %	0.1 %
I	0.63 A	1 μA	10^{-3} %	10^{-3} %
TC emf	836.0 μV	1 μV	0.1 %	0.9 %
Q	1.50 W	0.07 W	0.5 %	0.1 %
A	129.1 cm^2	1. cm^2	0.0 %	0.8 %
ΔX	2.61 cm	0.01 cm	0.0 %	0.3 %
ΔT	29.00 K	0.005 K	0.02 %	0.9 %
T	325.00 K	0.05 K	0.02 %	0.9 %
k	52.6 mW/m·K	0.25 mW/(m·K)	0.5 %	1.5 %

B. Measurements at 750 K (477°C)

(RH = 5.71 ohm; TC sensitivity: 38.4 μV/K)

Primary Variable	Value	Uncertainty	Imprecision	Bias
V	6.48 V	10 μV	10^{-3} %	0.1 %
I	1.14 A	10 μA	10^{-3} %	10^{-3} %
TC emf	15.9 mV	1 μV	0.01 %	0.02 %
Q	7.40 W	0.05 W	0.7 %	2 %
A	129.1 cm^2	1. cm^2	0.0 %	0.8 %
ΔX	2.61 cm	0.01 cm	0.0 %	0.3 %
ΔT	75.00 K	0.015 K	0.02 %	0.9 %
T	750.00 K	0.15 K	0.02 %	0.9 %
k	100.0 mW/m·K	0.7 mW/(m·K)	0.7 %	2.5 %

Appendix A: BASIC-Language Computer Program, "HT_GHP", for the High-Temperature Guarded-Hot-Plate Apparatus.

```
10    ! PROGRAM 'HT_GHP'            LAST REVISION DATE : 8-12-87
15    !     THIS PROGRAM OPERATES THE HIGH TEMPERATURE GUARDED HOT
20    ! PLATE THERMAL CONDUCTIVITY APPARATUS.
25    !     TEMPERATURES OF VARIOUS PARTS OF THE APPARATUS WILL
30    ! BE PRINTED OUT ON THE THERMAL PRINTER DURING THE SYSTEM'S
35    ! EQUILIBRATION PROCESS.  TC TEMPERATURE DATA WILL BE PRINTED OUT
40    ! AFTER THE DATA ACQUISITION PHASE IS COMPLETE AND WILL THEN BE
45    ! STORED ON FLOPPY DISK.
50    !    ABBREVIATIONS USED:  tc   =  thermocouple
55    !                        RTD   =  platinum resistance thermometer
60    !                         MH   =  MAIN (METERED) HEATER PLATE
65    !                         IG   =  INNER  GUARD   HEATER PLATE
70    !                         OG   =  OUTER  GUARD   HEATER PLATE
75    !                      T/BAH =  TOP / BOTTOM   HEATER PLATES
80    !    HPIB ADDRESS LIST:
85    !         PRINTER                  : 701
90    !         DVM 195 (IG)             : 706
95    !         DVM 195 (OG)             : 707
100   !         DVM 195 (TAHP)           : 708
105   !         DVM 195 (BAHP)           : 709
110   !         DVM 181 (MEAS. SENSORS)  : 710
115   !         DVM 181 (MHP)            : 712
120   !         MULTIPROGRAMMER          : 723
125   !    MULTIPROGRAMMER CARD ADDRESSES:
130   !         DIGITAL OUTPUT CARD      :  00
135   !         DIGITAL INPUT  CARD      :  01
140   !         HI SPEED A/D   CARD      :  02
145   !         RELAY OUTPUT   CARD      :  05
150   !         VOLTAGE  D/A   CARDS     :  07  ( METERED  AREA HEATER )
155   !                                  :  08  ( OUTER  GUARD  HEATER )
160   !                                  :  09  ( TOP AUXILIARY HEATER )
165   !                                  :  10  ( BOTTOM AUXILIARY HEATER )
170   !                                  :  11  ( INNER  GUARD  HEATER )
175   !         SCAN CONTROL   CARD      :  13
180   !         FET SCAN 16              :  15
185   !    PROGRAM SUBROUTINE LIST :
190    GOTO 1105      !    (SKIP FOLLOWING LIST TO BEGIN AT MAIN PROGRAM)
195    GOTO 10360     !    Adjust          (GOTO'S HERE ENABLE AUTOMATIC
200    GOTO 16915     !  * Atod_io          RENUMBERING)
205    GOTO 15280     !    Chan_switch
210    GOTO 17085     !    Data_read
215    GOTO 18505     !    Err_record
220    GOTO 7865      !    Final_averages
225    GOTO 12620     !    FNChan_sig
230    GOTO 9020      !    FNEmf_tc
235    GOTO 12525     !    FNOhms_rtd
240    GOTO 7275      !    FNTemp_rtd
245    GOTO 7445      !    FNTemp_tc
250    GOTO 13115     !    G_label
255    GOTO 6740      !    Init_run_vars
260    GOTO 17870     !    K_stor
265    GOTO 13745     !    K_ghp
270    GOTO 14480     !    Linear
275    GOTO 14690     !    Manual
280    GOTO 14395     !    Outseven
285    GOTO 10115     !    Pack_queue
290    GOTO 12215     !    Pblank
295    GOTO 11070     !    Pid
300    GOTO 15660     !    Plot_prep
305    GOTO 11760     !    Plot_switch
310    GOTO 10275     !    Poweroff
315    GOTO 10245     !    Poweron
320    GOTO 13255     !  * Read_io
325    GOTO 15410     !    Read_old_data
330    GOTO 9155      !    Record_data
335    GOTO 7690      !    Ref_rtd
340    GOTO 12740     !    Rescale_plot
345    GOTO 18110     !    Rtd_tune
350    GOTO 18075     !    Run_abort
355    GOTO 14090     !    Set_pnt_calc
360    GOTO 5375      !  * Sys_init
365    GOTO 5235      !    Sys_shutdown
```

```
370     GOTO 13515      !    Tc_store
375     GOTO 15545      !    Time_set
380     GOTO 16610      !    Update_plot
385     GOTO 14210      !  * Write_io
390    !
395    !  NOTE 1 : FILES PREFACED WITH  "*"  ARE FILES CONTAINING I/O
400    !  NOTE 2 : FOR ADDITIONAL VARIABLE DESCRIPTIONS SEE THE VARIABLE
405    !           DECLARATION SECTIONS IN THE MAIN PROGRAM AND IN SUB 'Sys_init'
410    !
415    !  PARAMETER  LIST:
420    !     SIMPLE VARIABLES:
425    !         Heater_sres : standard resistor in metered area heater line
430    !         Htr_res     : resistance of main heater
435    !         Htemp_lim   : high temperature limit on main heater plate
440    !         Ltemp_lim   : low  temperature limit on main heater plate
445    !         Nexp        : number of experiments
450    !         Ne          : array index counters for the Edat and Pdat arrays
455    !         Nr          : array index counter  for the Rtdat array
460    !         Nf          : array index counters for Kdat, Fedat, Ftdat, & Tme
465    !         Rtdpwr_sres : standard resistor in RTD current loop
470    !         Td0         : time relative to T0 when final data taking begins
475    !         T0          : absolute time (seconds) at start of controlled run
480    !         Ts          : time (seconds) between RTD data points
485    !         Tlim        : time limit (sec.) for system to reach equilibrium
490    !                    and take required data for calculation of conductivity
495    !         ***  ALSO SEE VARIABLE DECLARATION SECTION BELOW  ***
500    !     ARRAY VARIABLES :
505    !         Atune(4)    : Stores the tc temperatures used in phase 1 to
510    !                       calculate the RTD setpoint offsets in sub 'Rtd_tune'
515    !         Bad_inst(100) : Holding array for addresses of instruments that
520    !                         have had read errors
525    !         Bad_read_time$(100) : Stores the time at which instrument read
530    !                               errors have occurred
535    !         Cdata(5,2,3)  : Error/output array for 5 PID controllers
540    !                           ( IG, OG, TOP, BOTTOM and MAIN )
545    !                            over three time intervals
550    !         Cntrl_vlim(*) : High limit of the control voltage for each
555    !                         power supply
560    !         Cset(5,7)     : For the five PID controllers, respectively the
565    !                         set point (1), gain (2), integrator time (3),
570    !                         derivative time (4), integrator time bell width
575    !                         (5), gain bell width (6), and gain reduction
580    !                         factor (7)
585    !         Cwater(2,150) : Cold water inlet temperature record (temperature
590    !                         in degrees Celsius and time in seconds)
595    !         Edat(5,8000)  : CONTROL RTD-TEMP. (AND TC (VOLTS)) READINGS
600    !              ARRAY INDEX 1 holds IG tc READINGS (VOLTS)
605    !              ARRAY INDEX 2,3,4, AND 5 holds RTD TEMP. READINGS (C)
610    !                           (OG,TAP,BAP, AND MAIN PLATE RESPECTIVELY)
615    !         Fedat(8,Nfmax) : tc EMF (volts) data point storage (E AND delE)
620    !              ARRAY INDEX 1,3,5,and 7 hold EMF readings of 4 meas. tc's
625    !                           (TAP,TMP,BMP,BAP RESPECTIVELY)
630    !              ARRAY INDEX 2,4,6,and 8 hold EMF dev. of the 4 meas. tc's
635    !                           (TAP,TMP,BMP,BAP RESPECTIVELY)
640    !         Ftdat(8,Nfmax) : tc TEMP.(C) DATA POINT STORAGE
645    !              ARRAY INDEX 1,3,5,and 7 hold TEMP. of the 4 meas. tc's
650    !                           (TAP,TMP,BMP,BAP RESPECTIVELY)
655    !              ARRAY INDEX 2,4,6,and 8 hold Delta T's of the 4 meas. tc's
660    !                           (TAP,TMP,BMP,BAP RESPECTIVELY)
665    !         Fd(30) : Misc. data stored on disk file
670    !              ARRAY INDEX 01 - average run temperature
675    !              ARRAY INDEX 02 - thermal conductivity (W/(m*K))
680    !              ARRAY INDEX 03 - sample thickness (m)
685    !              ARRAY INDEX 04 - sample area density (kg/(m↑2))
690    !              ARRAY INDEX 05 - chamber gas pressure (mm Hg)
695    !              ARRAY INDEX 06 - emissivity of the plate material
700    !              ARRAY INDEX 07 - chamber gas code
705    !              ARRAY INDEX 08 - metered area diameter (m)
710    !              ARRAY INDEX 09 - average heater power (m)
715    !              ARRAY INDEX 10 - final heater resistance (ohms)
720    !              ARRAY INDEX 11 - Bot. aux. plate avg. T   (C)
725    !              ARRAY INDEX 12 - Bot. main plate avg. T   (C)
```

```
730    !              ARRAY INDEX 13 - Top  main plate avg. T   (C)
735    !              ARRAY INDEX 14 - Top  aux. plate avg. T   (C)
740    !              ARRAY INDEX 15 - Rc = code for run mode (1 = double sided;
745    !                         2 = single sided: top; 3 = single sided: bottom)
750    !              ARRAY INDEX 16 - Thi : avg. high temp. measured (C)
755    !              ARRAY INDEX 17 - Tlo : avg. low temp. measured (C)
760    !              ARRAY INDEX 18 - IG bottom (volts)
765    !              ARRAY INDEX 19 - IG combined (volts)
770    !              ARRAY INDEX 20 - IG top (volts)
775    !              ARRAY INDEX 21 - Dr  : IG gap width (m)
780    !              ARRAY INDEX 22 - Sc  : Plate spacer code
785    !                               (1 = quartz, 2 = stainless steel)
790    !              ARRAY INDEX 23 - Dxc : Corrected sample thickness (m)
795    !              ARRAY INDEX 24 - Ts  : Controller cycle time (seconds)
800    !              ARRAY INDEX 25 - Acor: Corrected main plate area (m↑2)
805    !              ARRAY INDEX 27 - Td0 : Start time of data aquisition (sec)
810    !              ARRAY INDEX 28 - Final averaging interval for data (sec)
815    !              ARRAY INDEX 29 - Starting point for final averaging interval
820    !              ARRAY INDEX 30 - Type of run code : 0 = normal or zero,
825    !                                 1 = OG offset, 2 = IG offset
830    !              ARRAY INDEX 26 - spare array space
835    !          File_num(Nexp) : file number for each experimental run
840    !          File_specs$(5)[80] : string array stored on disk file
845    !              ARRAY INDEX 1 HOLDS THE FILENAME, RUN DATE, AND OPER. MODE
850    !              ARRAY INDEX 2 HOLDS HI AND LO TEMP. SETPOINTS AND THE
855    !                            MAIN HEATER OPERATING MODE
860    !              ARRAY INDEX 3 HOLDS THE TIME TO EQUILIBRATION AND HOW IT
865    !                            WAS DETERMINED, AND TOTAL RUN TIME
870    !              ARRAY INDEX 4 CHAMBER GAS, SPECIMEN CODE (TOP AND BOTTOM),
875    !                            GUARD CODE (TOP AND BOTTOM)
880    !              ARRAY INDEX 5 MATERIAL DESCRIPTION/USER COMMENTS ON THE RUN
885    !          Htemp(Nexp) : High temp. for each experiment
890    !          Ht_mode(Nexp) : Main heater plate control mode indicator
895    !                1 = constant temperature mode
900    !                2 = constant power mode
905    !          Kdat(Nfmax)        : Data point store for 'run time' calc. of k
910    !          Last_reading(Indx) : Records last V on JRL channel 'Indx'
915    !          Ltemp(Nexp)        : Low temperature for each experiment
920    !          Mode$(4)[34]       : string vector for operating modes
925    !          Ok_flag(4)         : stability flag for RTD setpoint offsets
930    !          Op_mode(Nexp)      : Operation mode for each of the experiments
935    !                0 = double sided operation
940    !                1 = single sided operation (top)
945    !                2 = single sided operation (bottom)
950    !          Pdat(2,8000) : Heater power data storage array (these data are
955    !                         sampled every Ts seconds)
960    !              ARRAY INDEX 1 Main heater current (amps)
965    !              ARRAY INDEX 2 Main heater voltage (volts)
970    !          Queue(Nq)  : Holds scanner relay #'s (to be scanned in sequence)
975    !          Qseq1(Nqs1) : Holds the scanner sequence (s.c.) for the tc's
980    !          Qseq2(Nqs2) : Holds the s.c. for the iso-block temp. reading
985    !          Qseq3(Nqs3) : Holds the s.c. for the three IG parts
990    !          Qseq4(Nqs4) : Holds the s.c. for the main htr. current reading
995    !          Qseq5(Nqs5) : Holds the s.c. for the DVM zero reading updates
1000   !          Qseq6(Nqs6) : Holds the s.c. for phase 1 tc measurements
1005   !          Rtdat(2,Nrmax) : TEMP. (C) AND TIME (seconds) data point storage
1010   !                        for the isothermal reference block RTD
1015   !              ARRAY INDEX 1 Holds the temperature data
1020   !              ARRAY INDEX 2 Holds the time data
1025   !          Run_errors$(Err_max) : Record of errors occurring during a run
1030   !          Sdlim(5)   : 2 std.dev. limit for phase 1 RTD temp. stability
1035   !          Sp_corr(5) : RTD temp. setpoint corrections for phase 1 control
1040   !          Sp_errlim(5) : Temp. deviation from setpoint; limit for phase 1
1045   !                         stability.
1050   !          Splast(5,2): holds the previous and current RTD temp. setpoint
1055   !                        corrections.
1060   !          Tme(Nfmax) : TIME (sec) array for the data in Fedat,Ftdat,AND
1065   !                    Kdat  (one to one array index correspondence)
1070   !          Zhistory(3,2) : this holds the dvm zero values at the beginning
1075   !                         and end of phase 2
1080   !
1085   ! NOTE : FOR ADDITIONAL VARIABLE DESCRIPTIONS SEE THE VARIABLE DECLARATION
```

```
1090  !          SECTIONS IN THE MAIN PROGRAM AND SUB 'Sys_init'.
1095  !------------------------------------------------------------------------
1100  !       MAIN PROGRAM
1105   OPTION BASE 1
1110  ! COMMON DECLARATIONS
1115   COM /Adjustlocal/ Summ(5)
1120   COM /Adjloc2/ Last_reading(0:19)
1125   COM /Ioscan/ Queue(30),Nq,Qseq1(20),Nqs1,Qseq2(2),Nqs2,Qseq3(4),Nqs3,Qseq4(2),Nqs4
,Qseq5(2),Nqs5,Qseq6(9),Nqs6
1130   COM /Ctr1/ Cdata(5,2,3),Cset(5,7),Cntrl_vlim(5),Loop_label$(5)[14],Cstr$(7)[14],Mh
vmax
1135   COM /Constpwr/ Esum,Tsum,Tsum2,Etsum,P2n,Ntp2
1140   COM /Dr1/ Bad_curr
1145   COM /Dt1/ File_specs$(5)[80],Mode$(4)[34],Gas$(4)[10]
1150   COM /Fl/ Flag$[120]
1155   COM /Fld/ Disp_flag
1160   COM /Flgs/ Igflag,Prev_ne
1165   COM /Gr1/ Plot_view,Plot_type,Pindex
1170   COM /Gr2/ X1,X2,Xinc,Y1,Y2,Yinc,Xtit$[40],Ytit$[40]
1175   COM /Htr1/ Htr_res
1180   COM /Instr/ Mh181,Jrl181,Mh195,Tap195,Bap195,Ig195
1185   COM /Jrlchan/ Chan,Tchan,Dvm_cmmd$[40],Default_chan
1190   COM /Manual/ Powerflag,Vreading(5)
1195   COM /Mc1/ Ts,Ne,Edat(5,8000),Pdat(2,8000),Nr,Rtdat(2,2500),Nf,Fedat(8,250),Ftdat(8
,250),Kdat(250),Tme(250),Nrmax,Nfmax,Tlim
1200   COM /Mc2/ Heater_sres
1205   COM /Mc3/ Rtdpwr_sres
1210   COM /Mc5/ Op_mode(10),Htemp(10),Ltemp(10),Ht_mode(10),File_num(10),Set_temp(5)
1215   COM /Mc6/ Ntm,Ntr,Ntp,Ntz
1220   COM /Read1/ Io_error,Bad_instr(100),Bad_read_time$(100)[40]
1225   COM /Rn/ Run
1230   COM /Rtd_corr/ Tcorr_rtd,Rtd_adj_flag,Sp_corr(5)
1235   COM /Run_err/ Rterr,Run_errors$(100)[80],Err_max
1240   COM /Sb1/ T0,Td0
1245   COM /Sb2/ I_rtd,Tref,Emf_ref
1250   COM /Sb3/ Fd(30),Tavg_interval
1255   COM /Sdisp/ Screen_prnt
1260   COM /Stable/ Sdlim(5),Sp_errlim(5),Pnze,Ksd,Kslp,Knze
1265   COM /Tcst1/ Store_flag
1270   COM /Tune1/ Atune(4),Ok_flag(4),Splast(5,2)
1275   COM /Water/ Ncw,Cwater(2,150)
1280   COM /Zeros/ Zjrl181_200,Zjrl181_20,Zgap195,Zhistory(3,2)
1285  !    ARRAYS
1290   DIM Str1$[20],Str2$[20],Ans$[10],Ans2$[10],Lng_str$[160]
1295  !------------------------------------------------------------------------
1300  ! VARIABLE DECLARATIONS
1305  !    CONSTANTS
1310   Nq=30                   ! QUEUE ARRAY SIZE USED IN THE COMMON 'Ioscan'
1315   Nqs1=20                 !                     "
1320   Nqs2=2                  !                     "
1325   Nqs3=4                  !                     "
1330   Nqs4=2                  !                     "
1335   Nqs5=2                  !                     "
1340   Nqs6=9                  !                     "
1345   Default_chan=18         ! RTD CURRENT CHANNEL
1350   Screen_prnt=0           ! 1 = PRINTOUT TO CRT , 0 = NO PRINTOUT
1355   Ts=5                    ! CONTROLLER CYCLE TIME (seconds)
1360   Ntm=36                  ! # OF CYCLES BETWEEN tc MEASUREMENT READINGS
1365   Ntp=72                  ! # OF CYCLES BETWEEN POWER CURRENT READINGS
1370   Ntp2=240                ! # OF CYCLES USED IN SLOPE CALCULATION IN CONSTANT POWER MO
DE
1375   Ntr=24                  ! # OF CYCLES BETWEEN REFERENCE BLOCK READINGS
1380   Ntz=240                 ! # OF CYCLES BETWEEN EACH DVM ZERO READING
1385   Nrmax=2500              !  MAX. NUMBER OF REF. READINGS
1390   Nfmax=250               !  MAX. NUMBER OF tc DATA POINTS
1395   Tlim=7990.*Ts           !  11.1 HOURS ( ~8000 DATA POINTS @ 5 SEC/PT.)
1400   Tavg_interval=30*60     !  LENGTH OF THE DATA AVERAGING INTERVAL (SECONDS)
1405   Heater_sres=.0100008    !  SERIAL NUMBER 91023
1410   Rtdpwr_sres=99.994      !  SERIAL NUMBER 1550680
1415   Err_max=100             !  MAXIMUM NUMBER OF ERRORS TOLERATED DURING A RUN
1420   Htemp_lim=500
1425   Ltemp_lim=0
```

```
1430  !------------------------------------------------------------------------
1435  PRINTER IS 1
1440  !          SET THE DATE
1445  CALL Time_set
1450  !          INITIALIZE THE SYSTEM
1455  Flag$="OK"
1460  CALL Sys_init
1465  IF Flag$<>"OK" THEN Err_chk
1470  !------------------------------------------------------------------------
1475  !          GENERAL SYSTEM SPECIFICATION INITIALIZATION SECTION
1480  PRINT USING "@,15/"
1485  PRINT "    WOULD YOU LIKE TO READ THE GENERAL SPECIFICATIONS FOR THIS"
1490  PRINT "         RUN (OR SERIES OF RUNS) FROM A DISK FILE?"
1495  Ans$=""
1500  LINPUT "    (Y/N)?",Ans$
1505  IF Ans$<>"Y" AND Ans$<>"N" THEN GOTO 1500
1510  IF Ans$="Y" THEN CALL Read_old_data
1515  IF Fd(3)=0 THEN
1520     PRINT USING "@,15/"
1525     PRINT "  INPUT THE FOLLOWING GENERAL SPECIFICATIONS FOR THIS SET OF RUNS"
1530     Samp_dx=0
1535     INPUT " WHAT IS THE SAMPLE THICKNESS (cm)?",Samp_dx
1540     IF Fd(3)<=0 AND Samp_dx<=0 THEN GOTO 1535
1545     IF Samp_dx>0 THEN Fd(3)=Samp_dx/100. ! STORE PLATE THICKNESS IN METERS
1550     INPUT " WHAT IS THE SAMPLE AREA DENSITY (kg/m↑2)?",Fd(4)
1555     IF Fd(4)<=0 THEN GOTO 1550
1560     PRINT USING "@,5/"
1565     PRINT "    THE FOLLOWING GASES ARE USED IN THE CHAMBER :"
1570     FOR I=1 TO 4
1575        PRINT "              ";I;" : ";Gas$(I)
1580     NEXT I
1585     INPUT " INPUT THE NUMBER OF THE GAS USED IN THE CHAMBER ?",Gnum
1590     IF Gnum<1 OR Gnum>4 THEN GOTO 1585
1595     File_specs$(4)[1;20]=Gas$(Gnum)
1600     Fd(7)=Gnum
1605     PRINT USING "@"
1610     INPUT " WHAT IS THE CHAMBER GAS PRESSURE (mm Hg)?",Fd(5)
1615     IF Fd(5)<=0 THEN GOTO 1610
1620     PRINT USING "10/"
1625     PRINT "    THE DEFAULT VALUE OF THE EMISSIVITY IS :";Fd(6)
1630     INPUT " EMISSIVITY OF THE PLATE MATERIAL (JUST HIT 'ENTER' FOR THE DEFAULT) ?",
Fd(6)
1635     IF Fd(6)<=0 THEN GOTO 1630
1640     PRINT USING "@,10/"
1645     PRINT "      ENTER THE PLATE SPACER CODE :"
1650     PRINT "              1 = QUARTZ"
1655     PRINT "              2 = STAINLESS STEEL"
1660     INPUT " ENTER THE PLATE SPACER CODE NUMBER (1 OR 2)",Fd(22)
1665     IF Fd(22)<1 OR Fd(22)>2 THEN GOTO 1660
1670  END IF
1675  IF Fd(22)=1 THEN Psm$="QUARTZ"
1680  IF Fd(22)=2 THEN Psm$="STAINLESS STEEL"
1685  PRINT USING "@,10/,40A,/";"     GENERAL SPECIFICATIONS"
1690  PRINT "  SAMPLE THICKNESS (cm)         : ";Fd(3)*100
1695  PRINT "  SAMPLE AREA DENSITY (kg/m↑2): ";Fd(4)
1700  PRINT "  TYPE OF GAS                   : ";File_specs$(4)[1;20]
1705  PRINT "  CHAMBER GAS PRESSURE (mmHg) : ";Fd(5)
1710  PRINT "  EMISSIVITY OF THE PLATE      : ";Fd(6)
1715  PRINT "  PLATE SPACER MATERIAL        : ";Psm$
1720  PRINT USING "3/"
1725  Ans2$="N"
1730  INPUT " IS THIS INFORMATION CORRECT (Y/N)?",Ans2$
1735  IF Ans2$<>"Y" THEN
1740     PRINT " INPUT CORRECTIONS AS NECESSARY"
1745     GOTO 1535
1750  END IF
1755  PRINT USING "@,10/"
1760  PRINT "   THE MATERIAL DESCRIPTION/CURRENT USER COMMENT IS :"
1765  PRINT " "&""""&TRIM$(File_specs$(5))&""""
1770  LINPUT " IS THIS DESCRIPTION/COMMENT ACCEPTABLE (Y/N - DEFAULT IS Y)?",Ans$
1775  IF Ans$<>"Y" AND Ans$<>"N" AND Ans$<>"" THEN GOTO 1770
1780  IF Ans$="N" THEN
```

```
1785       LINPUT " ENTER YOUR MATERIAL DESCRIPTION/COMMENT (80 CHARACTER MAX.)",Lng_str$
1790       IF LEN(Lng_str$)>80 THEN
1795          DISP " YOU ARE TOO WORDY ... BE CONCISE (LESS THEN 80 CHARACTERS!)"
1800          WAIT 2
1805          GOTO 1785
1810       END IF
1815       File_specs$(5)=Lng_str$
1820    END IF
1825    File_specs$(4)[1]=File_specs$(4)&RPT$(" ",80-LEN(File_specs$(4)))
1830    IF LEN(TRIM$(File_specs$(4)))<21 THEN
1835       PRINT USING "@"
1840       LINPUT " ENTER THE TOP SPECIMEN CODE ( 10 CHAR. MAX. ) :",Lng_str$
1845       IF LEN(Lng_str$)>10 THEN GOTO 1840
1850       IF LEN(Lng_str$)>0 THEN File_specs$(4)[21,30]=Lng_str$
1855       LINPUT " ENTER THE TOP GUARD CODE ( 10 CHAR. MAX. ) :",Lng_str$
1860       IF LEN(Lng_str$)>10 THEN GOTO 1855
1865       IF LEN(Lng_str$)>0 THEN File_specs$(4)[41,50]=Lng_str$
1870       LINPUT " ENTER THE BOTTOM SPECIMEN CODE ( 10 CHAR. MAX. ) :",Lng_str$
1875       IF LEN(Lng_str$)>10 THEN GOTO 1870
1880       IF LEN(Lng_str$)>0 THEN File_specs$(4)[31,40]=Lng_str$
1885       LINPUT " ENTER THE BOTTOM GUARD CODE ( 10 CHAR. MAX. ) :",Lng_str$
1890       IF LEN(Lng_str$)>10 THEN GOTO 1885
1895       IF LEN(Lng_str$)>0 THEN File_specs$(4)[51,60]=Lng_str$
1900    END IF
1905    PRINT USING "@,5/"
1910    PRINT "    SPECIMEN AND SPECIMEN GUARD CODES :"
1915    PRINT USING "3/"
1920    PRINT "           TOP SPECIMEN CODE    : ";File_specs$(4)[21,30]
1925    PRINT "           TOP GUARD CODE       : ";File_specs$(4)[41,50]
1930    PRINT
1935    PRINT "           BOTTOM SPECIMEN CODE : ";File_specs$(4)[31,40]
1940    PRINT "           BOTTOM GUARD CODE    : ";File_specs$(4)[51,60]
1945    Ans$=""
1950    INPUT "   ARE THESE CODES CORRECT  (Y/N) ?",Ans$
1955    IF Ans$<>"Y" AND Ans$<>"N" THEN GOTO 1950
1960    IF Ans$="N" THEN GOTO 1835
1965    PRINT USING "@,10/"
1970    INPUT " HOW MANY RUNS DO YOU WANT TO PERFORM (<=10)?",Nexp
1975    IF Nexp>10 OR Nexp<1 THEN 1970
1980    PRINT USING "@,7/"
1985    File_specs$(1)[1]="____"
1990    INPUT " WHAT IS THE FILE NAME PREFIX (FOUR CHARACTERS)?",Ans$
1995    IF LEN(Ans$)>4 THEN
2000       PRINT USING "2/,50A,2/";"  ILLEGAL FILE NAME : IT IS TOOO LONG!!"
2005       GOTO 1985
2010    ELSE
2015       IF LEN(Ans$)<1 THEN GOTO 1985
2020       IF NUM(Ans$)<65 OR NUM(Ans$)>90 THEN
2025          PRINT USING "2/"
2030          PRINT " THE FIRST CHARACTER OF THE FILE SPECIFIER MUST BE A LETTER!"
2035          GOTO 1985
2040       END IF
2045       File_specs$(1)[1,LEN(Ans$)]=Ans$
2050    END IF
2055    PRINT USING "4/"
2060    PRINT " THE FILE NAME PREFIX FOR THIS SERIES OF RUNS WILL BE : ";File_specs$(1)
2065    Ans$=""
2070    INPUT " IS THIS PREFIX CORRECT (Y/N)?",Ans$
2075    IF Ans$<>"Y" THEN GOTO 1980
2080    Ans2$="N"
2085   ! IF THE RUN NUMBERS ARE SEQUENTIAL, INITIALIZE THE ENTIRE FILE NUM VECTOR
2090    IF Nexp>1 THEN
2095       Ans2$=""
2100       INPUT " WILL THE FILE RUN NUMBERS BE SEQUENTIAL (Y/N)?",Ans2$
2105       IF Ans2$="Y" THEN
2110          INPUT " WHAT IS THE RUN NUMBER OF THE FIRST RUN (0 TO 990)?",File_num(1)
2115          IF File_num(1)<0 OR File_num(1)>990 THEN
2120             DISP " ILLEGAL RUN NUMBER - TRY AGAIN "
2125             WAIT 3
2130             GOTO 2110
2135          END IF
2140          FOR I=1 TO Nexp-1
```

```
2145            File_num(I+1)=File_num(1)+I
2150         NEXT I
2155      ELSE
2160         IF Ans2$<>"N" THEN GOTO 2100
2165      END IF
2170   END IF
2175   !------------------------------------
2180   !   INDIVIDUAL RUN INITIALIZATION SECTION
2185   FOR Isis=1 TO Nexp
2190      IF Ans2$="N" THEN
2195         PRINT USING "20/"
2200         PRINT " WHAT IS THE FILE NUMBER FOR RUN NUMBER ";Isis;" (0 TO 990)?"
2205         INPUT " ",File_num(Isis)
2210         IF File_num(Isis)<0 OR File_num(Isis)>990 THEN
2215            DISP " FILE NUMBER OUT OF LEGAL RANGE !!"
2220            WAIT 3
2225            GOTO 2195
2230         END IF
2235      END IF
2240      PRINT
2245      PRINT " INPUT THE FOLLOWING PARAMETERS FOR RUN NUMBER ";Isis;""
2250      PRINT "                               ( FILE NUMBER : ";File_num(Isis);" )"
2255      PRINT USING "//,50A";"     A) OPERATING MODE "
2260      PRINT "              1 = DOUBLE SIDED OPERATION"
2265      PRINT "              2 = SINGLE SIDED OPERATION - TOP"
2270      PRINT "              3 = SINGLE SIDED OPERATION - BOTTOM"
2275      PRINT
2280      PRINT "     B) HIGH TEMPERATURE (C)  (";Ltemp_lim;" TO ";Htemp_lim;")"
2285      PRINT USING "/,28A,/";"     C) LOW TEMPERATURE (C) "
2290      PRINT "     D) MAIN HEATER PLATE CONTROL MODE"
2295      PRINT "              1 = CONSTANT TEMPERATURE MODE"
2300      PRINT USING "50A,/";"              2 = CONSTANT POWER MODE"
2305      INPUT "    INPUT THE MODE NUMBER (1,2,OR 3)",Op_mode(Isis)
2310      IF Op_mode(Isis)<>1 AND Op_mode(Isis)<>2 AND Op_mode(Isis)<>3 THEN 2305
2315      INPUT "    WHAT IS THE HIGH TEMPERATURE (C) ( MAIN HEATER PLATE TEMP. )?",Htemp
(Isis)
2320      IF Htemp(Isis)<Ltemp_lim OR Htemp(Isis)>Htemp_lim THEN
2325         PRINT "HIGH TEMPERATURE CHOICE IS OUT OF RANGE!"
2330         PRINT "THE RANGE IS ";Ltemp_lim;" TO ";Htemp_lim;" deg C"
2335         GOTO 2315
2340      END IF
2345      INPUT "    WHAT IS THE LOW TEMPERATURE (C) ?",Ltemp(Isis)
2350      IF Htemp(Isis)<Ltemp(Isis) THEN
2355         PRINT USING "//,60A";" THE LOW TEMP. VALUE IS HIGHER THAN THE HIGH TEMP. VAL
UE!"
2360         GOTO 2315
2365      END IF
2370      IF Ltemp(Isis)<(Ltemp_lim-10) THEN
2375         PRINT USING "//,60A";"  YOUR CHOICE OF LOW TEMPERATURE IS TOO LOW         "
2380         PRINT "        THE MINIMUM TEMPERATURE IS";Ltemp_lim;" DEG C"
2385         GOTO 2345
2390      END IF
2395      INPUT " INPUT MAIN HEATER PLATE CONTROL MODE (1=CONST TEMP.,2=CONST POWER)?",Ht
_mode(Isis)
2400      IF Ht_mode(Isis)<>1 AND Ht_mode(Isis)<>2 THEN GOTO 2395
2405      PRINT USING "@,/////,36A,D,6A";" THE FINAL PARAMETERS FOR RUN NUMBER ";Isis;" A
RE :"
2410      PRINT "                 ( FILE NUMBER";File_num(Isis);" )"
2415      PRINT
2420      PRINT "     SYSTEM";Mode$(4);Mode$(Op_mode(Isis))
2425      PRINT USING "27A,DDDD.DD";"     HIGH TEMPERATURE IS : ";Htemp(Isis)
2430      PRINT USING "27A,DDDD.DD";"     LOW  TEMPERATURE IS : ";Ltemp(Isis)
2435      SELECT Ht_mode(Isis)
2440      CASE 1
2445         PRINT "     MAIN HEATER PLATE CONTROL MODE : CONSTANT TEMPERATURE"
2450      CASE 2
2455         PRINT "     MAIN HEATER PLATE CONTROL MODE : CONSTANT POWER"
2460      END SELECT
2465      Ans$="N"
2470      INPUT " ARE THESE VALUES ACCEPTABLE (Y/N)?",Ans$
2475      IF Ans$<>"Y" THEN GOTO 2245
2480   NEXT Isis
```

```
2485   PRINT USING "@,10/"
2490   High_temp=MAX(Htemp(*))
2495   Highr=FNOhms_rtd(5,High_temp)
2500   Fluke_cur=INT((.180/Highr)*10000.)/10.
2505   IF Fluke_cur>1.5 THEN Fluke_cur=1.5
2510   IF Fluke_cur<.2 THEN Fluke_cur=.2
2515   PRINT "   THE OPTIMUM CURRENT FOR THE FLUKE POWER SUPPLY IS : ";Fluke_cur;" millia
mps"
2520   PRINT "          (THE ACCEPTABLE RANGE IS .2 TO ";Fluke_cur;" mA)"
2525   PRINT
2530   PRINT "   HIT 'CONTINUE' WHEN POWER SUPPLY HAS BEEN SET."
2535   PAUSE
2540   PRINT USING "@,10/"
2545   PRINT "      WHEN YOU ARE READY TO BEGIN THE COMPUTER CONTROLLED"
2550   PRINT "      RUN ( OR SET OF RUNS ) HIT 'CONTINUE'"
2555   PAUSE
2560   PRINT USING "@"
2565  !         END OF USER INTERACTION INITIALIZATION SECTION
2570  !----------------------------------------------------------------------
2575  !    MAIN PROGRAM LOOP
2580   FOR Run=1 TO Nexp
2585      CALL Init_run_vars(Run)
2590  !----------------------------------------------------------------------
2595  !  THIS SECTION BRINGS THE GUARDED HOT PLATE APPARATUS TO
2600  !  THERMAL EQUILIBRIUM.  IT THEN CHECKS THE STABILITY OF THE
2605  !  MAIN HEATER POWER SUPPLY.
2610  !  AFTER THE SYSTEM IS STABLE, READINGS ARE TAKEN FROM THE MEASUREMENT
2615  !  THERMOCOUPLES UNTIL ENOUGH DATA HAS BEEN TAKEN.
2620      IF Flag$<>"OK" THEN Err_chk
2625      GRAPHICS ON
2630  !   ALLOCATE ARRAYS USED TO TEST THE SYSTEM FOR EQUILIBRIUM
2635      Tst_size=INT(180/Ts)+1
2640      Ktst_size=INT(Tavg_interval/(Ts*Ntm))+1
2645      IF Tst_size>36 THEN Tst_size=36
2650      ALLOCATE X(1:Tst_size),Y(1:Tst_size),Sdset(1:5),Spdev_set(1:5),Tcalm(1:5)
2655      Ptst_sz=48
2660      ALLOCATE Xp(1:Ptst_sz),Yp(1:Ptst_sz),Xk(1:Ktst_size),Yk(1:Ktst_size)
2665      ON ERROR GOTO Sysequil_err
2670  !----------------------------------------------------------------------
2675  !   PRINT HEADER FOR EQUIL. TEMPERATURE AND CONTROL OUTPUT
2680      PRINTER IS 701
2685      PRINT USING "@,2/"
2690      PRINT " FILE NAME : ";File_specs$(1)[1,7]
2695      PRINT "     ";File_specs$(1)[11,80]
2700      IF Ht_mode(Run)=1 THEN
2705         Mhcd$="TEMP."
2710      ELSE
2715         Mhcd$="POWER"
2720      END IF
2725      PRINT " HIGH TEMP.(C)=";Htemp(Run);" ; LOW TEMP.(C)=";Ltemp(Run);" ; MAIN HEATE
R CONTROL : CONSTANT ";Mhcd$
2730      PRINT USING "2/"
2735      PRINT "   TIME      MHP TEMP      OG TEMP     TAHP TEMP    BAHP TEMP    IG DIFF
  "
2740      PRINT "   (min)        (C)           (C)           (C)          (C)          (microV
)     "
2745      PRINT
2750      PRINTER IS 1
2755      PRINT USING "@"
2760  !----------------------------------------------------------------------
2765  ! DEFINE THE REQUIRED SOFTKEYS
2770      FOR I=0 TO 9
2775         ON KEY I LABEL "" GOSUB 4990     !  (DUMMY RETURN)
2780      NEXT I
2785      ON KEY 4 LABEL "  ABORT RUN",15 CALL Run_abort
2790      ON KEY 2 LABEL "HIT IF STABLE" GOSUB 4990
2795      ON KEY 12 LABEL "HIT IF STABLE" GOSUB Set_td_vars
2800      ON KEY 7 LABEL " FINISH RUN" GOSUB 4990
2805      ON KEY 17 LABEL " FINISH RUN" GOSUB Set_fini_vars
2810      ON KEY 0 LABEL " CHANGE PLOT" CALL Plot_switch
2815      ON KEY 3 LABEL " ALPHA TOGGLE" GOSUB Alpha_toggle
2820      ON KEY 5 LABEL "  PID ADJUST " CALL Manual
```

```
2825      IF Screen_prnt=1 THEN
2830         ON KEY 8 LABEL "     ON" GOSUB 4990
2835      ELSE
2840         ON KEY 8 LABEL "     OFF" GOSUB 4990
2845      END IF
2850  !------------------------------------------------------------------------
2855  ! TAKE INITIAL DATA FOR THE TEMPERATURE REFERENCE BLOCK
2860      T0=TIMEDATE+5       ! THE TRUE VALUE OF T0 WILL BE TAKEN IN ~5 SEC.
2865  !            DETERMINE THE RTD CURRENT
2870      CALL Chan_switch(18,"R3B1X")       !    RTD CURRENT STD. RES. VOLTAGE
2875      IF Screen_prnt THEN PRINT "   ISO. BLOCK CURRENT MEASUREMENT"
2880      WAIT 5
2885      CALL Read_ia(V,Timee,Jr1181,"R3B1X",I_rtd*Rtdpwr_sres)
2890      I_rtd=V/Rtdpwr_sres
2895      IF I_rtd>(Fluke_cur/1000.) OR I_rtd<.0002 THEN Flag$="   RTD CURRENT SETTING IS
 OUT OF ACCEPTABLE RANGE!!"
2900      Last_reading(18)=V
2905      IF Flag$<>"OK" THEN GOTO Err_chk
2910  !            DETERMINE THE ISO-BLOCK INITIAL TEMPERATURE
2915      CALL Chan_switch(1,"R3B1X")        !    RTD VOLTAGE (ISO.-BLOCK)
2920      IF Screen_prnt THEN PRINT "   ISO. BLOCK VOLTAGE MEASUREMENT"
2925      WAIT 5
2930      CALL Read_ia(V,Timee,Jr1181,"R3B1X",I_rtd*Rtdpwr_sres*1.08)
2935      Last_reading(1)=V
2940      CALL Ref_rtd(V,Timee)
2945      IF Flag$<>"OK" THEN GOTO Err_chk
2950  !            DETERMINE INITIAL VALUE OF MAIN HEATER RESISTANCE
2955      CALL Outseven(.01)              ! SET A TICKLER VOLTAGE
2960      CALL Chan_switch(19,"R3B1X")  ! VOLTAGE ACROSS MAIN HTR. LINE STD.RES.
2965      IF Screen_prnt THEN PRINT "   HEATER RESISTANCE (INITIAL MEASUREMENT)"
2970      WAIT 5
2975      CALL Read_ia(Vi,Timee,Jr1181,"R3B1X",.0001)
2980      CALL Read_ia(Vp,Timee,Mh195,"R0X",.01)
2985      Htr_res=Heater_sres*Vp/Vi
2990      IF Htr_res<2.5 OR Htr_res>6.5 THEN Flag$=" MAIN HEATER RESISTANCE MEASUREMENT E
RROR"
2995      IF Screen_prnt THEN PRINT "     INITIAL MAIN HEATER RESISTANCE = ";Htr_res;" OH
MS"
3000      CALL Outseven(0)
3005  !            SET THE JRL RELAY TO A DEFAULT CHANNEL
3010      CALL Chan_switch(Default_chan,"R3B1X")
3015      WAIT 1
3020  !------------------------------------------------------------------------
3025  !            INITIALIZE LOOP COUNTERS AND SET ADJUST CYCLE
3030      Td0=0              !  TIME OF FINAL DATA AQUISITION (RELATIVE TO T0)
3035      Igflag=0           !  END OF RUN FLAG TO MEASURE 3 IG tc's
3040      Store_flag=0       !  tc STORAGE FLAG (0=NOT READING DATA FOR A SET)
3045      User_override=0 !  COMPUTER/USER CONTROL FLAG FOR DATA STORAGE
3050      Comp_fini$="N"    !  COMPUTER SIGNAL FLAG ('Y' WHEN IT THINKS IS DONE)
3055      Fini$="N"         !  SET TO 'Y' WHEN DATA TAKING DONE OR TLIM EXCEEDED
3060      Stable$="N"       !  OVERALL STABILITY FLAG (TAKE FINAL DATA : PHASE 2)
3065      T_stable$="N"     !  TEMPERATURE STABILITY FLAG
3070      P_stable$="N"     !  POWER (VOLTAGE) STABILITY FLAG
3075      Check=0           !  K STABILITY FLAG
3080      Got_it=0          !  DATA RECORDING FLAG
3085      T0=TIMEDATE       !  ZERO TIME READING FOR RUN
3090      Rtd_adj_flag=0    !  RTD RUNTIME CALIBRATION FLAG
3095      Bad_curr=0        !  BAD RTD CURRENT READ COUNTER
3100      ON CYCLE Ts,10 CALL Adjust
3105  !------------------------------------------------------------------------
3110  !  EQUILIBRIUM TEST AND USER INTERACT LOOP ( PERFORMED BETWEEN CALLS TO
3115  !  THE CONTROLLER SUB 'Adjust' )
3120      REPEAT
3125  !         EXECUTE EVERYTHING WITHIN THIS IF BLOCK ONCE PER CONTROL CYCLE
3130         IF Queue(1)>0 THEN CALL Data_read
3135         IF Ne<>Nelast THEN
3140            Nelast=Ne
3145            IF Nf<>0 THEN
3150               IF Tme(Nf)<>0 AND Rtdat(2,Nr)>Tme(Nf) AND Kdat(Nf)=0 THEN CALL K_stor
3155            END IF
3160            CALL Update_plot("Y")
3165  !               CHECK THE RTD TEMPERATURES UNTIL STABILITY IS REACHED
```

```
3170  !                  DO LIN. REG. TEST  (USE EDATA ARRAY FROM POINT Ne
3175  !                  BACKWARDS ON EACH RTD DATA SET - TIME & TEMP )
3180          IF Ne>Tst_size+1 AND T_stable$="N" AND Stable$="N" THEN
3185             Net=Ne
3190             FOR I=1 TO 5
3195                FOR J=1 TO Tst_size
3200                   X(J)=(Net-Tst_size+J)*Ts    !    X-AXIS (SEC.)
3205                   Y(J)=Edat(I,Net-Tst_size+J) !    Y-AXIS (degC)
3210                   IF I=1 THEN Y(J)=Y(J)*1.E+6 !    Y-AXIS (microV)
3215                NEXT J
3220                CALL Linear(X(*),Y(*),Tst_size,D1,Slpe,Mean,Stddev)
3225                Spdev_set(I)=Mean-(Set_temp(I)-Sp_corr(I))     ! S.P. DEV.
3230                Sdset(I)=2*Stddev                  ! 2 STD. DEV. FROM THE MEAN
3235                IF ((ABS(Spdev_set(I)))<Sp_errlim(I)) AND (Sdset(I)<Sdlim(I)) AND R
td_adj_flag AND Tcalm(I)=0 THEN
3240                   Tcalm(I)=1
3245                   OUTPUT 701;"                    "&Loop_label$(I)&" TEMPERATURE IS S
TABLE."
3250                END IF
3255                Ln=INT(10000.*Sdset(I))/10000.
3260                IF Screen_prnt THEN PRINT " LOOP#";I;" DEV. FROM S.P.=";INT((Spdev_
set(I))*10000.)/10000.;" STD.DEV. =";Ln
3265             NEXT I
3270             IF SUM(Tcalm)=5 THEN
3275                T_stable$="Y"
3280                PRINT USING "@"
3285                PRINT "             ALL TEMPERATURE LOOPS ARE STABLE."
3290             END IF
3295             IF TIMEDATE-Tcorr_rtd>5*60 AND Rtd_adj_flag=0 THEN
3300                CALL Pack_queue(Queue(*),Nq,Qseq6(*),Nqs6)
3305                Tcorr_rtd=TIMEDATE
3310             END IF
3315          END IF
3320  !------------------------------------------------------------------------------
3325  !     AFTER 50 POWER READINGS HAVE BEEN TAKEN, TEST FOR STABILITY USING THE
3330  !     THE LAST Ptst_sz=48 DATA POINTS
3335          IF Ne>50 AND P_stable$="N" AND (Tcalm(5)=1 OR T_stable$="Y") AND Stable$=
"N" THEN
3340             Net=Ne
3345             FOR I=1 TO Ptst_sz
3350                Xp(I)=Ts*(Net-Ptst_sz+I)
3355                Yp(I)=Pdat(2,Net-Ptst_sz+I)*Pdat(1,Net-Ptst_sz+I)
3360             NEXT I
3365             CALL Linear(Xp(*),Yp(*),Ptst_sz,D1,Slpe,Mean,Stddev)
3370  !       USE THE TIME/POWER CURVE SLOPE, % STD. DEV. , AND POWER NOISE
3375  !       LIMIT ASSESS MAIN HEATER POWER STABILITY.
3380             Pnoise=2*Stddev
3385             IF Screen_prnt THEN
3390                PRINT "       2 STD.DEV. OF POWER : PNOISE (W)=";INT(Pnoise*1000.)/
1000

3395             END IF
3400             IF Pnoise<Pnze AND Rtd_adj_flag THEN
3405                P_stable$="Y"
3410                OUTPUT 701;"                    MAIN HEATER POWER IS STABLE"
3415             END IF
3420          END IF
3425  !------------------------------------------------------------------------------
3430  !                  IF THE TEMP. LOOPS AND MAIN HEATER POWER ARE STABLE THEN
3435  !                  BEGIN PHASE 2   (OR IF TIME LIMIT EXCEEDED; 5 Hrs.)
3440  !GOTO 3395 !!!      TO SKIP OVER TRANSITION TO PHASE 2 FOR TUNING OF PROGRAM
3445          IF ((T_stable$="Y" AND P_stable$="Y") OR TIMEDATE-T0>5*3600.) AND Stable$
="N" THEN
3450             Stable$="Y"
3455             IF T_stable$="Y" AND P_stable$="Y" THEN
3460                File_specs$(3)[1]="COMPUTER DETERMINED STABILITY AT "&VAL$(INT((TIM
EDATE-T0)/60))&" min. ;  "
3465             ELSE
3470                File_specs$(3)[1]="PHASE 1 TIME LIMIT EXCEEDED AT "&VAL$(INT((TIMED
ATE-T0)/60))&" min. ;  "
3475                OUTPUT 701;" PHASE 1 TIME LIMIT EXCEEDED AT "&VAL$(INT((TIMEDATE-T0
)/60))&" min.  "
3480             END IF
```

```
3485               Td0=Ne*Ts       ! TIME AT WHICH WE BEGAN TAKING FINAL DATA
3490                               !     ( RELATIVE TO T0 )
3495               P2n=Ne
3500               Esum=0
3505               Tsum=0
3510               Tsum2=0
3515               Etsum=0
3520               BEEP 83*5,.2
3525               PRINT USING "@"
3530             END IF
3535             IF Stable$="Y" THEN
3540               ON KEY 2 LABEL "" GOSUB 4990
3545   !          ********  COMPUTER DATA TEST SECTION ON k  *********
3550   !                    IF DATA IS OK SET   Check = 1
3555               IF Nf>Ktst_size+1 AND Ne*Ts-Td0>3600 THEN
3560                 Nft=Nf
3565                 IF Store_flag<>0 THEN Nft=Nf-1
3570                 FOR J=1 TO Ktst_size
3575                   Xk(J)=Tme(J+Nf-Ktst_size)          !  X-AXIS (SEC.)
3580                   Yk(J)=Kdat(J+Nf-Ktst_size)         !  Y-AXIS (mW/M*K)
3585                 NEXT J
3590                 CALL Linear(Xk(*),Yk(*),Ktst_size,D1,Slpe,Mean,Stddev)
3595                 IF Mean<>0 THEN
3600                   Kstddev=2*Stddev                   !  2 STD DEV (SDEV)
3605                   Kslope=Slpe*3600                   !  k CHANGE PER HOUR
3610                 END IF
3615                 Net=Ne
3620                 FOR I=1 TO Ptst_sz                   !  FIND SDEV OF POWER
3625                   Xp(I)=Ts*(Net-Ptst_sz+I)
3630                   Yp(I)=Pdat(2,Net-Ptst_sz+I)*Pdat(1,Net-Ptst_sz+I)
3635                 NEXT I
3640                 CALL Linear(Xp(*),Yp(*),Ptst_sz,D1,Slpe,Mean,Stddev)
3645                 Qnoise=2*Stddev                      !  POWER NOISE
3650                 IF ABS(Kslope)<Kslp AND ((Kstddev<Ksd) OR (Qnoise<Knze)) THEN Check
=1
3655                 Ln=INT(1000*Qnoise)/1000
3660                 IF Screen_prnt THEN
3665                   PRINT "   KSLOPE=";INT(Kslope*1000)/1000;"  2SD=";INT(Kstddev*10
00)/1000;"  NOISE=";Ln
3670                 END IF
3675               END IF
3680               IF Nf>=INT((Tavg_interval)/(Ts*Ntm)+1) AND Check=1 AND Comp_fini$<>"Y"
 THEN
3685                 Comp_fini$="Y"
3690                 Tcomputer=TIMEDATE
3695               END IF
3700               ON KEY 1 LABEL " FINISH MODE" GOSUB Finish_mode
3705               IF User_override THEN
3710                 ON KEY 6 LABEL "    MANUAL" GOSUB 5090
3715               ELSE
3720                 ON KEY 6 LABEL "   COMPUTER" GOSUB 5090
3725               END IF
3730             END IF
3735             IF Comp_fini$="Y" AND Fini$="N" THEN
3740   !                USER OVERRIDE SECTION FOR RUN TERMINATION
3745   !               (IF NO RESPONSE IN 30 MIN.,THEN STORE THE DATA)
3750               IF NOT User_override THEN
3755                 IF TIMEDATE-Tcomputer>1800 THEN Fini$="Y"
3760                 IF Ne MOD 6=0 AND Queue(1)=-1 THEN BEEP 83*3,.5
3765                 IF Queue(1)=-1 THEN
3770                   DISP " THE COMPUTER HAS A DATA SET IT WILL STORE (AT T-0) UNLESS
 THE USER OVERRIDES"
3775                 END IF
3780                 Disp_flag=TIMEDATE
3785                 ON KEY 8 LABEL " DELAY CNTDWN" GOSUB Count_delay
3790                 Tcont=INT(((1800-(TIMEDATE-Tcomputer))/60)*10)/10
3795                 Tcount$=VAL$(Tcont)
3800                 Tcount$="T-"&Tcount$&" MINUTES"
3805                 ON KEY 3 LABEL Tcount$ GOSUB 5090
3810               ELSE
3815                 DISP ""
3820                 ON KEY 8 LABEL "" GOSUB 5090
```

```
3825                ON KEY 3 LABEL "" GOSUB 5090
3830             END IF
3835             IF Tcant<=0 THEN GOSUB Set_fini_vars
3840          END IF
3845          IF Fini$="Y" AND Igflag=0 THEN
3850             Igflag=1
3855  !      ADD THE IG SCAN CHANNELS TO THE QUEUE
3860             CALL Pack_queue(Queue(*),Nq,Qseq3(*),Nqs3)
3865          END IF
3870          IF Queue(1)=-1 AND Chan<>Default_chan THEN
3875             CALL Chan_switch(Default_chan,"R3B1X")
3880          END IF
3885       END IF
3890       IF TIMEDATE-Disp_flag>10 AND Queue(1)=-1 THEN DISP ""
3895       IF (TIMEDATE-T0>Tlim OR Nf=Nfmax) AND Fini$="N" THEN GOSUB Set_fini_vars
3900    UNTIL (Fini$="Y" AND Queue(1)<0) OR Flag$<>"OK"
3905    OFF CYCLE
3910    CALL Chan_switch(Default_chan,"R3B1X")
3915    DEALLOCATE X(*),Y(*),Sdset(*),Spdev_set(*),Xp(*),Yp(*),Tcalm(*),Xk(*),Yk(*)
3920    IF Nf>0 THEN
3925       WHILE Kdat(Nf)=0 AND Nf>1
3930          Nf=Nf-1
3935       END WHILE
3940    END IF
3945  !--------------------------------------------------------------------------
3950    IF Set_temp(2)<>(Htemp(Run)+Ltemp(Run))/2 THEN Fd(30)=1
3955    IF Set_temp(1)<>0 THEN Fd(30)=2
3960    IF Stable$="N" THEN File_specs$(3)[1]="SYSTEM STABILITY NOT ACHIEVED ;  "
3965    File_specs$(3)[1]=File_specs$(3)[1;LEN(File_specs$(3))]&"TOTAL RUN TIME : "&VAL$(INT((TIMEDATE-T0)/60))&" min."
3970  !             PRINTOUT HEADER AND FDAT ARRAY (tc TEMPERATURES VS. TIME)
3975    IF Nf>0 THEN           !  IF THERE IS FINAL DATA...PRINT IT OUT
3980    ON ERROR GOTO Prntr_err
3985       PRINTER IS 701
3990       PRINT USING "@,/,16A,8A,/,8X,72A,2/";"    FILE NAME : ",File_specs$(1)[1,7],File_specs$(3)
3995       PRINT
4000       PRINT "          TABLE OF CORRECTED THERMOCOUPLE TEMPERATURES :"
4005       PRINT
4010       PRINT "TIME  TOP AUX. PLATE   TOP MAIN PLATE   BOT. MAIN PLATE   BOT. AUX. PLATE  POWER"
4015       PRINT "(min)   T(C)  del T      T(C)  del T      T(C)  del T       T(C)  del T   (Watt)"
4020       PRINT
4025       I=1
4030       WHILE Kdat(I)<>0
4035          PRINT USING "X,DDD,1X,#";Tme(I)/60.
4040          FOR J=2 TO 8 STEP 2
4045             IF J=8 THEN PRINT USING "X,#"
4050             IF (Ftdat(J-1,I)+Ftdat(J,I))<10000. THEN
4055                PRINT USING "DDDD.DDD,1X,#";Ftdat(J-1,I)+Ftdat(J,I)
4060             ELSE
4065                PRINT USING "5A,4X,#";">10↑4"
4070             END IF
4075             Deltat=Ftdat(J,I)
4080             IF ABS(Deltat)<10 THEN
4085                PRINT USING "SD.DDD,1X,#";Deltat
4090             ELSE
4095                PRINT USING "4A,3X,#";">"&VAL$(SGN(Deltat))&"10 "
4100             END IF
4105             IF J<>8 THEN PRINT USING "1X,#"
4110          NEXT J
4115          Nepp=INT(Tme(I)/Ts)
4120          IF Nepp>Ne OR Nepp<1 THEN
4125             IF Nepp>Ne THEN Nepp=Ne
4130             IF Nepp<1 THEN Nepp=1
4135             PRINT " TIME/DATA POINT ERROR FOR POWER CALC. AT ";Tme(I)/60;" MINUTES"
4140             CALL Err_record(" TIME/DATA POINT ERROR FOR POWER CALC. AT "&VAL$(Tme(I)/60)&" MINUTES")
4145             BEEP 83*3,.1
4150          END IF
```

```
4155            Power=Pdat(1,Nepp)*Pdat(2,Nepp)
4160            IF Power<1000 THEN
4165               PRINT USING "DDD.DD";Power
4170            ELSE
4175               PRINT USING "1X,5A";">1000"
4180            END IF
4185            I=I+1
4190         END WHILE
4195         PRINT
4200         PRINT "  DVM ZERO OFFSETS (microV) AT BEGINNING AND END OF PHASE 2 :"
4205         PRINT "    GAP195      : BEGIN=";Zhistory(3,2)*1.E+6;" , END=";Zhistory(3,1)*1.E+6
4210         PRINT "    JRL181 (R3) : BEGIN=";Zhistory(1,2)*1.E+6;" , END=";Zhistory(1,1)*1.E+6
4215         PRINT "    JRL181 (R2) : BEGIN=";Zhistory(2,2)*1.E+6;" , END=";Zhistory(2,1)*1.E+6
4220         PRINT
4225         PRINT "  PLATE TEMPERATURE CONTROL SETPOINT OFFSETS ( T (tc) - T (RTD) ) :"
4230         PRINT "          TOP PLATE = ";Sp_corr(3);" K"
4235         PRINT "         MAIN PLATE = ";Sp_corr(5);" K"
4240         PRINT "       BOTTOM PLATE = ";Sp_corr(4);" K"
4245         PRINT USING "3/"
4250         PRINT "    INLET WATER TEMPERATURE HISTORY :"
4255         PRINT
4260         ALLOCATE Cwdum(1:2*Ncw)
4265         FOR Jay=1 TO Ncw
4270            Cwdum(Jay*2)=Cwater(1,Jay)        !     TEMPERATURE
4275            Cwdum(Jay*2-1)=Cwater(2,Jay)      !         TIME
4280         NEXT Jay
4285         PRINT Cwdum(*)
4290         DEALLOCATE Cwdum(*)
4295         IF Rterr<>0 THEN
4300            PRINT USING "2/"
4305            PRINT "        ERRORS THAT OCCURRED DURING PROGRAM EXECUTION :"
4310            FOR Isis=1 TO Rterr
4315               PRINT "    "&VAL$(Isis)&" : "&Run_errors$(Isis)
4320            NEXT Isis
4325         END IF          !  +++
4330         PRINTER IS 1
4335         ON ERROR GOTO Sysequil_err
4340      END IF
4345   !-------------------------------------------------------------------------
4350 !       THIS SECTION GIVES THE USER 10 MIN. TO DISPOSE ANY PLOTS
4355      PRINT USING "@"
4360      FOR I=10 TO 20
4365         BEEP 83*I,.01
4370      NEXT I
4375      BEEP 83*21,.3
4380      FOR I=1 TO 9
4385         ON KEY I LABEL "" GOTO 4420
4390      NEXT I
4395      DISP ""
4400      ON KEY 4 LABEL "   END RUN  " GOTO 4455
4405      ON KEY 0 LABEL " CHANGE PLOT" GOTO 4415
4410      GOTO 4420
4415      CALL Plot_switch
4420      Tpause=TIMEDATE
4425      LOOP
4430         Tbusy=TIMEDATE-Tpause
4435         DISP " YOU HAVE ";10-(INT((TIMEDATE-Tpause)/6))/10;" MIN. TO DISPOSE ANY PLOTS YOU WANT."
4440         IF TIMEDATE-Tpause-Tbusy>2 THEN Tpause=TIMEDATE
4445      EXIT IF TIMEDATE-Tpause>600        !  EXIT LOOP AFTER 10 MIN.
4450      END LOOP
4455      DISP ""
4460   !-------------------------------------------------------------------------
4465      IF (Flag$="OK" OR Flag$="RUN ABORTED") AND Nf>0 THEN CALL Final_averages
4470      IF Nf>0 AND Got_it=0 THEN
4475         CALL Record_data
4480         Got_it=1
4485      END IF
4490      IF Flag$<>"OK" THEN Err_chk
```

```
4495   NEXT Run
4500  !      END OF MAIN PROGRAM LOOP
4505  !!!
4510  !----------------------------------------------------------------------------
4515  !!!
4520   IF Flag$="OK" AND ERRM$="" THEN
4525      CALL Sys_shutdown
4530      GOTO End
4535   END IF
4540 Err_chk:  !
4545  !    FATAL ERROR SECTION
4550   OFF CYCLE
4555   PRINTER IS 1
4560   CALL Sys_shutdown            !  ZERO OUT THE POWER SIGNALS
4565   IF Nf>0 AND Got_it=0 THEN
4570      CALL Record_data
4575      Got_it=1
4580   END IF
4585   IF Flag$="RUN ABORTED" THEN
4590      PRINT USING "@,10/"
4595      PRINT "    THE RUN HAS BEEN ABORTED!"
4600      GOTO End
4605   END IF
4610   IF Flag$<>"OK" THEN
4615      PRINT USING "@,5/"
4620      PRINT " THE FOLLOWING PROBLEM OR PROBLEM AREA CAUSED PROGRAM TERMINATION :"
4625      PRINT Flag$
4630      PRINT
4635   END IF
4640   IF ERRM$<>"" THEN
4645      PRINT " THE FOLLOWING ERROR WAS THE LAST TO OCCUR DURING PROGRAM EXECUTION :"
4650      PRINT ERRM$
4655   END IF
4660  !     END OF ERROR CHECKING SECTION
4665   IF Flag$<>"OK" AND Run<=Nexp THEN
4670      ON KBD GOTO Rezume
4675      DISP "STRIKE ANY KEY TO END THE BEEPING"
4680      FOR I=14 TO 28 STEP 2
4685         BEEP 81.38*I,.01
4690      NEXT I
4695      BEEP 81.38*28,.4
4700      Tbeep=TIMEDATE
4705      IF TIMEDATE-Tbeep<.4 THEN GOTO 4705
4710      GOTO 4680
4715 Rezume:     !
4720      OFF KBD
4725      DISP " "
4730   END IF
4735   GOTO End
4740  !----------------------------------------------------------------------------
4745 Prntr_err: !   ERROR TRAP SECTIONS FOR MAIN LOOP
4750  !         TRAP PRINTER ERRORS AND RECOVER PROGRAM OPERATION
4755    PRINTER IS 1
4760    PRINT "    ";ERRM$
4765    BEEP 83*10,.3
4770    ON ERROR GOTO Sysequil_err
4775    GOTO 4330
4780 Sysequil_err: !  ERROR TRAP SECTION FOR MAIN LOOP
4785   ON ERROR GOTO Sysequil_err
4790   IF ERRL(2680) OR ERRL(2690) OR ERRL(2695) OR ERRL(2725) OR ERRL(2730) OR ERRL(2735
) OR ERRL(2740) OR ERRL(2745) THEN
4795      PRINTER IS 1
4800      PRINT USING "@,5/"
4805      PRINT "     PRINTER ERROR ... CORRECT THE PROBLEM IF POSSIBLE"
4810      PRINT
4815      PRINT "           ";ERRM$
4820      BEEP 83*15,.3
4825      GOTO 2750
4830   END IF
4835   IF ERRL(3245) THEN GOTO 3250
4840   IF ERRL(3410) THEN GOTO 3415
4845   Errl_flag=0
```

```
4850   IF ERRL(3475) THEN GOTO 3480
4855  !  IF THE ERROR IS NOT A PLOTTER ERROR THEN SET 'FLAG$' AND STOP EXECUTION
4860   IF Flag$="OK" THEN Flag$=" MAIN LOOP ERROR "&ERRM$
4865   PRINTER IS 1
4870   GOTO Err_chk
4875  !-----------------------------------------------------------
4880 Alpha_toggle: !
4885   IF Screen_prnt=1 THEN
4890      Screen_prnt=0
4895      PRINT USING "@"
4900      ON KEY 8 LABEL "    OFF" GOSUB 4990
4905   ELSE
4910      Screen_prnt=1
4915      ON KEY 8 LABEL "     ON" GOSUB 4990
4920   END IF
4925   RETURN
4930  !-----------------------------------------------------------
4935 Set_td_vars: !
4940   Stable$="Y"
4945   P_stable$="Y"
4950   T_stable$="Y"
4955   File_specs$(3)[1]=" USER DETERMINED STABILITY AT : "&VAL$(INT((Ne*Ts)/6+.5)/10)&"
min. ;  "
4960   Td0=Ne*Ts
4965   P2n=Ne
4970   Esum=0
4975   Tsum=0
4980   Tsum2=0
4985   Etsum=0
4990   RETURN
4995  !-----------------------------------------------------------
5000 Set_fini_vars: !
5005  !             COLLECT DATA FOR AT LEAST 'Tavg_interval' SECONDS BEFORE STORAGE
5010   IF (Nf>INT((Tavg_interval)/(Ntm*Ts)+1)) OR (TIMEDATE-T0>Tlim) OR (Nf=Nfmax) THEN
5015      Fini$="Y"
5020      DISP " THE TERMINATION SEQUENCE HAS BEGUN ... DO NOT HIT ANY OTHER KEYS!!"
5025      ON KEY 7 LABEL "FINISHING RUN!" GOSUB 5090
5030      CALL Pack_queue(Queue(*),Nq,Qseq3(*),Nqs3)
5035      Igflag=1
5040   ELSE
5045      IF Td0>0 THEN
5050         T_to_wait=INT((Tavg_interval-(TIMEDATE-T0)+Td0+3*Ts*Ntm)/6)/10
5055         IF T_to_wait<.1 THEN T_to_wait=.1
5060         DISP " YOU DON'T HAVE ENOUGH DATA!   YOU NEED ";T_to_wait;" MORE MINUTES OF
DATA"
5065      ELSE
5070         DISP " YOU DON'T HAVE ANY DATA YET"
5075      END IF
5080   END IF
5085   Disp_flag=TIMEDATE
5090   RETURN
5095  !-----------------------------------------------------------
5100 Count_delay: !
5105   Tcomputer=TIMEDATE
5110   RETURN
5115  !-----------------------------------------------------------
5120 Finish_mode: !
5125   IF User_override=1 THEN
5130      User_override=0
5135      Tcomputer=TIMEDATE
5140   ELSE
5145      User_override=1
5150   END IF
5155   RETURN
5160  !---------------------------------------------------------------------------------------
5165 End:             !      END OF THE MAIN PROGRAM
5170   PRINT USING "2/,50A";"                                END OF THE PROGRAM"
5175   IF Io_error>0 THEN
5180      PRINT
5185      PRINT "     IO ERRORS OCCURED DURING PROGRAM EXECUTION , THEY"
5190      PRINT "     ARE LISTED IN THE ARRAYS :    BAD_INSTR(*)"
5195      PRINT "                                AND    BAD_READ_TIME$(*)"
```

```
5200      PRINT
5205   END IF
5210   PAUSE
5215   END
5220   !!!
5225   !!!
5230  ! ///////////////////////////////////////////////////////////////////////
5235   SUB Sys_shutdown
5240  !    TURN OFF ALL POWER SUPPLIES
5245      ON TIMEOUT 7,1 GOTO Shutdown_err
5250      ON ERROR GOTO Shutdown_err
5255      PRINT
5260      FOR I=7 TO 11
5265         Write_io(723,"OP,"&VAL$(I)&",0,T")
5270      NEXT I
5275      CALL Outseven(0)
5280      GRAPHICS OFF
5285      PRINTER IS 1
5290      PRINT USING "@,3/,30A";" ALL POWER SUPPLIES OFF"
5295  !   OPEN ALL JULIE RELAY CONTACTS
5300      Write_io(723,"OP,0,255T")
5305      CALL Chan_switch(2,"R3B1X")
5310      PRINT USING "50A,/";"  JULIE RELAY SET TO REST STATE : CHANNEL 2 SHORT"
5315      SUBEXIT
5320 Shutdown_err: !
5325      IF ERRL(5265) THEN
5330         IF Flag$="OK" THEN Flag$="SHUTDOWN_ERR - POWER SUPPLY CONTROL PROBLEM"
5335         GOTO 5300
5340      END IF
5345      IF ERRL(5300) THEN
5350         IF Flag$="OK" THEN Flag$="SHUTDOWN_ERR - JULIE RELAY CONTROL PROBLEM"
5355      END IF
5360      IF Flag$="OK" THEN Flag$="  ERROR IN 'Sys_shutdown' :"&ERRM$
5365   SUBEND
5370  ! ///////////////////////////////////////////////////////////////////////
5375   SUB Sys_init
5380  !   THIS SUBROUTINE PROMPTS THE OPERATOR TO TURN ON ALL EQUIPMENT
5385  !   AND WAIT A SPECIFIED LENGTH OF TIME (1 hr) FOR THE SYSTEM TO
5390  !   WARM UP.  AFTER THE WARMUP PERIOD IS COMPLETE THIS ROUTINE :
5395  !            1) SETS ALL DVM'S TO INITIAL SETTINGS
5400  !            2) ZERO'S OUT D/A SIGNALS TO THE KEPCO POWER SUPPLIES
5405  !       AND  3) INITIALIZES ALL CONSTANT VALUE DATA ARRAYS
5410  !   VARIABLES :
5415  !      Alpha(5) : Alpha values for the RTD calibrations
5420  !      Delta(5) : Delta values for the RTD calibrations
5425  !      R0(5) : RTD resistance values at 0 deg C
5430  !      Tc_data(9) : NISIL-NICROSIL T.C. COEFFICIENTS FOR THE BASE
5435  !                   EQUATION OF EMF AS A FUNCTION OF TEMPERATURE.
5440  !                   ( from NBS monograph 161, p.49, table 7.3.2 )
5445  !      Tc_correction(4) : COEFFICIENTS FOR ADJUSTMENT OF NISIL-NICRO.
5450  !                         TC BASE EQUATION TO FIT EXPERIMENTAL DATA FOR
5455  !                         THE TC WIRE SPOOLS USED.
5460  !                        ( curve fit to (experimental data - predicted data)
5465  !                        June 28,1984 NBS calibration of tc wire and NBS
5470  !                        monograph 161, p.133, table C2 )
5475  !      Quartic_data(5,2) : COEFFICIENTS FOR NISIL-NICROSIL TC BASE
5480  !                          EQUATION OF TEMP. AS A FUNCTION OF EMF (USED AS
5485  !                          A FIRST APPROX. OF TEMP. IN A NEWTON-RAPHSON
5490  !                          ITERATION).
5495  !                          ( from NBS monograph 161, p.106, table A9 )
5500      COM /Adjustlocal/ Summ(*)
5505      COM /Dt1/ File_specs$(*),Mode$(*),Gas$(*)
5510      COM /Conv_dat/ R0(1:5),Alpha(1:5),Delta(1:5)
5515      COM /Ctr1/ Cdata(*),Cset(*),Cntrl_vlim(*),Loop_label$(*),Cstr$(*),Mhvmax
5520      COM /Fl/ Flag$
5525      COM /Htr1/ Htr_res
5530      COM /Htr3/ Htrlabel$(1:5)[14]
5535      COM /Instr/ Mh181,Jrl181,Mh195,Tap195,Bap195,Ig195
5540      COM /Ioscan/ Queue(*),Nq,Qseq1(*),Nqs1,Qseq2(*),Nqs2,Qseq3(*),Nqs3,Qseq4(*),Nqs
4,Qseq5(*),Nqs5,Qseq6(*),Nqs6
5545      COM /Jrlchan/ Chan,Tchan,Dvm_cmmd$,Default_chan
5550      COM /Read1/ Io_error,Bad_instr(*),Bad_read_time$(*)
```

```
5555      COM /Rtd_corr/ Tcorr_rtd,Rtd_adj_flag,Sp_corr(*)
5560      COM /Sb3/ Fd(*),Tavg_scan
5565      COM /Stable/ Sdlim(*),Sp_errlim(*),Pnze,Ksd,Kslp,Knze
5570      COM /Tcdat/ Tc_data(1:9),Tc_correction(1:4),Quartic_data(1:5,1:2)
5575      COM /Zeros/ Zjrl181_200,Zjrl181_20,Zgap195,Zhistory(*)
5580      ALLOCATE Heater_res(1:5)
5585      Scuz=-1
5590 !       ZERO OUT THE PID SUMM ARRAY
5595      MAT Summ= (0)
5600 !       ZERO OUT PART OF THE SPECIFICATIONS STRING ARRAY
5605      File_specs$(1)=RPT$(" ",80)
5610      File_specs$(2)=RPT$(" ",80)
5615      File_specs$(3)=RPT$(" ",80)
5620 !       DISPLAY SYSTEM POWER UP EQUIPMENT LIST AND TIME LIMIT
5625      T0_warmup=TIMEDATE
5630      PRINT USING "@,/"
5635      PRINT CHR$(129)
5640      PRINT "                                                                   "
5645      PRINT "          TURN ON THE FOLLOWING PIECES OF EQUIPMENT :              "
5650      PRINT "                                                                   "
5655      PRINT "                A) THE PRINTER                                     "
5660      PRINT "                B) THE MULTIPROGRAMMER                             "
5665      PRINT "                C) THE SIX KEITHLEY DVM'S  ( 4-195'S AND 2 181'S )  "
5670      PRINT "                D) THE FIVE KEPCO POWER SUPPLIES                   "
5675      PRINT "                E) BOTH JULIE RELAYS                               "
5680      PRINT "                F) THE FLUKE POWER SUPPLY ( BETWEEN 1.5mA AND 0.5mA) "
5685      PRINT "                G) THE   WATER  SUPPLY !!                          "
5690      PRINT ""
5695      PRINT "                                                                   "
5700      PRINT "        WHEN THE EQUIPMENT HAS BEEN TURNED ON, WAIT 1 HR.          "
5705      PRINT "        FOR THE SYSTEM TO COMPLETELY WARM UP.                      "
5710      PRINT "                                                                   "
5715      PRINT CHR$(128)
5720      PRINT USING "9X,1A,46A";CHR$(131)," WHEN THE SYSTEM IS READY, HIT   ' CONTINUE
' "
5725      PRINT CHR$(128)
5730      PAUSE
5735      PRINT USING "@,5/"
5740         GOTO 5800      !!!  REMOVABLE SKIP  !!!
5745      IF (TIMEDATE-T0_warmup)<3600 THEN
5750         INPUT " HAS IT REALLY BEEN ON FOR ONE HOUR (Y/N)??",Ans$
5755         IF Ans$="Y" THEN
5760            PRINT "   OK ...  I WAS JUST CHECKING!"
5765            WAIT 1.5
5770         ELSE
5775            PRINT "    JUST AS I SUSPECTED!! ... YOU MUST BE PATIENT! "
5780            WAIT 1.5
5785            GOTO 5630
5790         END IF
5795      END IF
5800 !    ASSIGN DVM ADDRESSES
5805       Ig195=706              ! INNER GUARD DVM
5810       Mh195=707              ! MAIN HEATER VOLTAGE DVM
5815      Tap195=708              ! TOP AUX. PLATE DVM
5820      Bap195=709              ! BOTTOM AUX. PLATE DVM
5825      Jrl181=710              ! JRL DVM
5830       Mh181=712              ! MAIN HEATER DVM
5835 !-------------------------------------------
5840 !    SET THE RANGE AND DISPLAY VALUES OF ALL THE DVM'S
5845      PRINT USING "@,5/"
5850      PRINT "          DVM'S ARE NOW BEING SET "
5855      ON TIMEOUT 7,1.0 GOTO Init_err
5860      FOR J=1 TO 2            ! SEND OUT THE DVM SET COMMANDS TWICE
5865         Scuz=0
5870         FOR I=706 TO 709
5875            OUTPUT I;"R0X"   ! SET THE 195's TO THE AUTO RANGING MODE
5880            WAIT .2
5885            ENTER I;Dum
5890            WAIT .2
5895         NEXT I
5900         Scuz=1
5905         OUTPUT Mh181;"R3B1X"   !  MAIN RTD
```

```
5910        WAIT .2
5915        ENTER Mh181;Dum
5920        WAIT 2
5925        OUTPUT Jrl181;"R3B1X"
5930        WAIT .2
5935        ENTER Jrl181;Dum
5940        WAIT .2
5945     NEXT J
5950  !  ZERO OUT ALL MULTIPROGRAMMER D/A SIGNALS TO THE KEPCO POWER SUPPLIES
5955     Scuz=2
5960     OUTPUT 723;"OP,7,0,8,0,9,0,10,0,11,0T"
5965  !     ZERO OUT RELAY CARD SIGNAL TO THE 8 BIT D/A FOR THE MAIN POWER S.
5970     CALL Outseven(0)
5975  !      CYCLE THE JULIE RELAYS
5980     PRINT "          THE JULIE RELAYS ARE NOW BEING CYCLED"
5985     FOR I=0 TO 19
5990        Jrl_com=FNChan_sig(I)
5995        OUTPUT 723;"OP,0,"&VAL$(Jrl_com)&"T"
6000        WAIT .05
6005     NEXT I
6010  !        TAKE ZERO READINGS FOR THE JRL181 DVM (ACROSS A SHORT)
6015     PRINT "          ZERO READINGS ARE BEING TAKEN OVER SEVERAL DVM RANGES"
6020     PRINT "              ( THIS TAKES APPROXIMATELY 1 MINUTE )"
6025     Scuz=2
6030     Jrl_com=FNChan_sig(2)
6035     OUTPUT 723;"OP,0,"&VAL$(Jrl_com)&"T"
6040     Scuz=1
6045     Write_io(Jrl181,"R3B1X")
6050     PRINT "         JRL DVM 181 200 mV SCALE ZERO READING"
6055     WAIT 15
6060     ENTER Jrl181;Zjrl181_200
6065     PRINT "                   ZERO READING IS : ";Zjrl181_200*1.E+6;" microvolts"
6070     Write_io(Jrl181,"R2B1X")
6075     PRINT "         JRL DVM 181 20 mV SCALE ZERO READING"
6080     WAIT 15
6085     ENTER Jrl181;Zjrl181_20
6090     PRINT "                   ZERO READING IS : ";Zjrl181_20*1.E+6;" microvolts"
6095     CALL Chan_switch(8,"R3B1X")     !  SET JRL RELAY TO TOTAL GAP SIGNAL
6100     PRINT "         GAP DVM 195  ZERO READING/CALCULATION"
6105     WAIT 15
6110     ENTER Jrl181;V01
6115     Scuz=0
6120     ENTER Ig195;V02
6125     Zgap195=V02-(V01-Zjrl181_200)
6130     PRINT "                   ZERO READING IS : ";Zgap195*1.E+6;" microvolts"
6135  !        SET THE JRL RELAY ON THE DEFAULT CHANNEL
6140     CALL Chan_switch(Default_chan,"R3B1X")   ! SET JRL RELAY TO DEFAULT
6145     Scuz=-1
6150  !---------------------------------------------
6155     PRINT "          THE CONSTANT DATA ARRAYS ARE NOW BEING INITIALIZED"
6160  !  LOAD THE CONSTANT DATA ARRAYS
6165     MAT Cdata= (0.)
6170     MAT Bad_read_time$= ("")
6175     MAT Bad_instr= (0)
6180     MAT Sp_corr= (0)
6185     Io_error=0
6190  ! LOAD OPERATING MODE STRING ARRAY
6195     Mode$(1)=" DOUBLE SIDED "
6200     Mode$(2)=" SINGLE SIDED - TOP "
6205     Mode$(3)=" SINGLE SIDED - BOTTOM "
6210     Mode$(4)=" OPERATING MODE :"
6215  !     LOAD CHAMBER GAS LIST
6220     DATA "HELIUM","NITROGEN","AIR","ARGON"
6225     READ Gas$(*)
6230  !      LOAD HEATER LABEL ARRAY
6235     DATA "GAP(IG) HTR.","OG HTR.","TOP AUX. HTR.","BOT. AUX. HTR.","MAIN HEATER"
6240     READ Htrlabel$(*)
6245  !      LOAD STRING ARRAY FOR CONTROLLER ADJUST SOFTKEY LABELS
6250     DATA "SETPOINT ","  GAIN (K)","  Ti (SEC)","  Td (SEC)","Ti BELL WIDTH"," K BEL
L WIDTH","K REDUC. FACT."
6255     READ Cstr$(*)      !     CONTROLLER PARAMETERS
6260     DATA " INNER GUARD"," OUTER GUARD","TOP AUX PLATE","BOT. AUX PLATE"," MAIN PLAT
```

E"
```
6265      READ Loop_label$(*)
6270   ! ADD CONSTANT DATA TO THE FD(*) ARRAY AND CANCEL PORTIONS OF IT
6275      Fd(1)=0.
6280      Fd(2)=0.
6285      FOR I=9 TO 20
6290         Fd(I)=0.
6295      NEXT I
6300      FOR I=23 TO 30
6305         Fd(I)=0.
6310      NEXT I
6315      Fd(24)=Ts                ! CONTROL CYCLE TIME
6320      Fd(6)=.85                !   PLATE EMISSIVITY
6325      Fd(8)=.1254              ! METERED AREA DIAMETER (m)
6330      Fd(21)=.0024             ! GAP WIDTH (m)
6335   !  RESISTANCE VALUES OF THE PLATE HEATERS (USED TO CALC. Cntrl_vlim)
6340   !     NOTE: IF THE TOTAL RESISTANCE IS >11 THE CONTROL V-LIM. WILL BE 10v
6345      Heater_res(1)=10.2+.9  !  INNER GUARD HEATER RESISTANCE + LEAD RES.
6350      Heater_res(2)=11.9     !  OUTER GUARD HEATER RESISTANCE
6355      Heater_res(3)=14.3     !  TOP AUX. PLATE HEATER RESISTANCE
6360      Heater_res(4)=14.3     !  BOTTOM AUX. PLATE HEATER RESISTANCE
6365      Heater_res(5)=3.7+.9   !  MAIN PLATE HEATER RESISTANCE + LEAD RES.
6370      Htr_res=Heater_res(5)  ! MAIN PLATE HTR. RES.(RE-MEASURED DURING RUN)
6375   !  CALCULATE THE CONTROL VOLTAGE HIGH LIMITS FROM THE HEATER PLATE RES.
6380   !  ( BASED ON  5A - 55V MAX. OUTPUT FOR POWER SUPPLY AND A  0 TO 10 VOLT
6385   !  CONTROLLER SIGNAL )
6390      FOR I=1 TO 5
6395         Cntrl_vlim(I)=Heater_res(I)*5./(55./10.)
6400         IF Cntrl_vlim(I)>10. THEN Cntrl_vlim(I)=10.
6405         IF I=5 THEN Mhvmax=Heater_res(I)*5.  ! M.H. v MAX.(BASED ON 5A)
6410      NEXT I
6415   !      ASSIGN VALUES OF STD. DEV. LIMIT AND SET POINT DEV. LIMIT FOR
6420   !      STABILITY CRITERIA ON ALL PLATES ( TEMP. CONTROLLED LOOPS ).
6425      DATA 1.0,1.0               !  INNER GUARD     (MICROVOLTS; LINE 3145)
6430      DATA 2.0,0.2               !  OUTER GUARD
6435      DATA .01,.005              !  TOP AUX. PLATE
6440      DATA .01,.005              !  BOTTOM AUX. PLATE
6445      DATA .002,.001             !  MAIN PLATE
6450      FOR I=1 TO 5
6455         READ Sdlim(I),Sp_errlim(I)
6460      NEXT I
6465   !      ASSIGN VALUES OF SLOPE AND 2 STD. DEV. FOR POWER
6470   !      AND 2 STD.DEV. AND SLOPE FOR k , AND 2 STD. DEV. FOR POWER
6475   !      THE MAIN PLATE POWER AND THERMAL CONDUCTIVITY
6480      DATA .003                  !  POWER
6485      READ Pnze
6490      DATA 0.3,0.08,.0020        !  THERMAL CONDUCTIVITY (k)
6495      READ Ksd,Kslp,Knze
6500   ! DATA FOR THE RTD SERIAL NUMBERS Z603,J57,Z597,Z602,AND J52,RESPECTIVELY
6505   !      1=REF. BLOCK, 2=OG, 3=TOP PLATE, 4=BOTTOM PLATE, 5=MAIN PLATE
6510      DATA 100.059,99.63,100.269,100.049,99.977
6515      READ R0(*)
6520      R0(3)=R0(3)-.86     !!!  SYSTEM DEPENDENT 'TUNING' FACTOR (TAHP)
6525      R0(4)=R0(4)+.052    !!!  SYSTEM DEPENDENT 'TUNING' FACTOR (BAHP)
6530      R0(5)=R0(5)-.31     !!!  SYSTEM DEPENDENT 'TUNING' FACTOR (MAIN PLATE)
6535      DATA .00389613,.00390382,.00390608,.00390002,.00390316
6540      READ Alpha(*)
6545      DATA 1.310780,1.501776,1.456476,1.398146,1.510977
6550      READ Delta(*)
6555   !  DATA FOR NIS.-NICRO. T.C. TEMP./VOLT CONVERSION EQUATION
6560   !       NBS MONOGRAPH 161 , PAGE 49 , TABLE 7.3.2 FOR AWG 14 WIRE
6565      DATA 25.897798582,1.6656127713E-2,3.1234962101E-5,-1.7248130773E-7
6570      DATA 3.6526665920E-10,-4.4390833504E-13,3.1553382729E-16
6575      DATA -1.2150879468E-19,1.9557197559E-23
6580      READ Tc_data(*)
6585      DATA 2.73134350E2,2.94724845E2,3.87205729E-2,3.21321378E-2
6590      DATA -1.09710024E-6,-2.89538382E-7,5.25218480E-11,5.02114728E-12
6595      DATA -1.14636136E-15,-2.61445196E-17
6600      READ Quartic_data(*)
6605      DATA -4.06943153E-1,3.14764457E-1,-1.07467859E-3,1.29379614E-6
6610      READ Tc_correction(*)
6615    !   LOAD THE SCANNER QUEUES
```

```
6620      DATA 18,1,3,13,4,14,5,15,6,16,16,6,15,5,14,4,13,3,18,1
6625      READ Qseq1(*)    !  tc CHANNELS FOR T AND DEL T OF THE PLATES
6630                       !  (AND REF. RTD CHANNELS AFTER Td0 IS ASSIGNED).
6635      DATA 18,1
6640      READ Qseq2(*)    !  REF. RTD CHANNELS BEFORE Td0 IS ASSIGNED.
6645      DATA 7,8,9,2
6650      READ Qseq3(*)    !  IG tc SEQUENCE (TAKEN AT END OF RUN).
6655      DATA 19,11
6660      READ Qseq4(*)    !  MAIN HEATER CURRENT READING.
6665      DATA 2,8
6670      READ Qseq5(*)    !  DVM 'ZERO' CHANNELS (JRL181 AND IG195)
6675      DATA 2,3,13,4,14,5,15,6,16
6680      READ Qseq6(*)    !  T.C. READ FOR RTD CALIBRATION IN PHASE 1
6685     !     PAUSE FOR A FEW SECONDS TO ALLOW USER TO VIEW SCREEN
6690      WAIT 3
6695      SUBEXIT
6700 Init_err:  !
6705      Flag$="SYSTEM INITIALIZATION ERROR"
6710      IF Scuz=0 THEN Flag$="SYSTEM INITIALIZATION ERROR, KEITHLEY 195 PROBLEM"
6715      IF Scuz=1 THEN Flag$="SYSTEM INITIALIZATION ERROR, KEITHLEY 181 PROBLEM"
6720      IF Scuz=2 THEN Flag$="SYSTEM INITIALIZATION ERROR, MULTIPROGRAMMER PROBLEM"
6725      BEEP 81.38*12,1.5
6730   SUBEND
6735  ! ////////////////////////////////////////////////////////////////////////
6740   SUB Init_run_vars(N)
6745  !  THIS ROUTINE INITIALIZES VARIABLE ARRAYS THAT MUST BE RE-INITIALIZED
6750  !  AT THE BEGINNING OF EACH RUN.
6755  !  PARAMETERS :
6760  !         N : CURRENT RUN NUMBER
6765  !
6770      ALLOCATE Dumm$[80],Dumm4$[80],Dumm5$[80]
6775      COM /Ioscan/ Queue(*),Nq,Qseq1(*),Nqs1,Qseq2(*),Nqs2,Qseq3(*),Nqs3,Qseq4(*),Nqs
4,Qseq5(*),Nqs5,Qseq6(*),Nqs6
6780      COM /Ctr1/ Cdata(*),Cset(*),Cntrl_vlim(*),Loop_label$(*),Cstr$(*),Mhvmax
6785      COM /Manual/ Powerflag,Vreading(*)
6790      COM /Mc1/ Ts,Ne,Edat(*),Pdat(*),Nr,Rtdat(*),Nf,Fedat(*),Ftdat(*),Kdat(*),Tme(*)
,Nrmax,Nfmax,Tlim
6795      COM /Mc5/ Op_mode(*),Htemp(*),Ltemp(*),Ht_mode(*),File_num(*),Set_temp(*)
6800      COM /Dt1/ File_specs$(*),Mode$(*),Gas$(*)
6805      COM /Fl/ Flag$
6810      COM /Gr1/ Plot_view,Plot_type,Pindex
6815      COM /Gr2/ X1,X2,Xinc,Y1,Y2,Yinc,Xtit$,Ytit$
6820      COM /Run_err/ Rterr,Run_errors$(*),Err_max
6825      COM /Sb3/ Fd(*),Tavg_interval
6830      COM /Tcst1/ Store_flag
6835      COM /Tune1/ Atune(*),Ok_flag(*),Splast(*)
6840      COM /Water/ Ncw,Cwater(*)
6845  ! ZERO THE DATA ARRAYS
6850      MAT Edat= (0.)
6855      MAT Ftdat= (0.)
6860      MAT Pdat= (0.)
6865      MAT Rtdat= (0.)
6870      MAT Fedat= (0.)
6875      MAT Kdat= (0.)
6880      MAT Tme= (0.)
6885      MAT Queue= (-1)
6890      MAT Ok_flag= (0)
6895      MAT Cwater= (0)
6900      MAT Run_errors$= ("")
6905      Rterr=0
6910      Ncw=0
6915      Ne=0
6920      Nr=0
6925      Nf=0
6930  !         MISC. VARIABLE INITIALIZATIONS
6935      Store_flag=0        !  THIS IS THE BEGIN STORE SEQUENCE FLAG ON tc's
6940  !   SET THE POWER ON/OFF FLAG
6945      Powerflag=1         !  1=ON ; 0=OFF
6950  !   SET FILE_SPECS ARRAY TO NULL STRINGS
6955      Dumm$=File_specs$(1)[1,4]
6960      Dumm4$=File_specs$(4)
6965      Dumm5$=File_specs$(5)
```

```
6970      MAT File_specs$= (RPT$(" ",80))
6975   ! FILL THE File_specs$ ARRAY WITH REFERENCE DATA ON THE CURRENT FILE
6980      File_specs$(4)[1]=Dumm4$
6985      File_specs$(5)[1]=Dumm5$
6990      File_specs$(1)[1]=Dumm$
6995      Dumm$=VAL$(File_num(N))
7000      SELECT LEN(Dumm$)
7005      CASE 1
7010         Dumm$="00"&Dumm$
7015      CASE 2
7020         Dumm$="0"&Dumm$
7025      CASE ELSE
7030      END SELECT
7035   !     ZERO OUT PART OF THE FD(*) ARRAY
7040      Fd(1)=0.
7045      Fd(2)=0.
7050      FOR I=9 TO 20
7055         Fd(I)=0.
7060      NEXT I
7065      FOR I=23 TO 30
7070         Fd(I)=0.
7075      NEXT I
7080      Fd(24)=Ts
7085      Fd(15)=Op_mode(N)
7090      Dumm$=Dumm$&"   "&DATE$(TIMEDATE)&"  "&TIME$(TIMEDATE)
7095      File_specs$(1)[5]=Dumm$&"   "&Mode$(4)&Mode$(Op_mode(N))
7100   ! ASSIGN THE CONTROLLER TEMPERATURE SET POINTS (C)
7105      Set_temp(5)=Htemp(N)               !   MAIN HEATER TEMPERATURE SETPOINT
7110      Set_temp(2)=(Htemp(N)+Ltemp(N))/2  !   OUTER GUARD TEMPERATURE SETPOINT
7115      M=Op_mode(N)       ! 1=DBLE SIDED, 2=TOP, 3=BOTTOM
7120      IF M=1 OR M=2 THEN Set_temp(3)=Ltemp(N)  !     TOP PLATE S.P.
7125      IF M=1 OR M=3 THEN Set_temp(4)=Ltemp(N)  ! BOTTOM PLATE S.P.
7130      IF M=2 THEN Set_temp(4)=Htemp(N)         ! BOTTOM PLATE S.P.
7135      IF M=3 THEN Set_temp(3)=Htemp(N)         !     TOP PLATE S.P.
7140      Set_temp(1)=0.                           ! IG Delta T SETPOINT
7145      CALL Set_pnt_calc                        ! CALCULATE SETPOINTS IN OHMS
7150      IF N=1 THEN
7155   ! ASSIGN VALUES OF K (GAIN), Ti, Td, Ti BELL WIDTH (TiBW), K BELL WIDTH
7160   ! (KBW), AND K REDUCTION FACTOR (KRF) FOR EACH CONTROL LOOP ON FIRST RUN.
7165   !
7170   !!!
7175   !      (VALUES FOR TiBW AND Kbw ARE IN VOLT UNITS FOR IG)
7180         DATA 180E3, 90, 0, 7E-5, 6E-8, .33   ! controller #1     IG
7185         DATA  200, 400, 0,  1.0, .002,  .1   ! controller #2     OG
7190         DATA  600, 200, 0, 0.14, 1E-4, .038  ! controller #3     TOP PLATE
7195         DATA  600, 200, 0, 0.14, 1E-4, .038  ! controller #4  BOTTOM PLATE
7200         DATA  100, 4000, 0, 0.04, .007, .05  ! controller #5    MAIN PLATE
7205         FOR I=1 TO 5
7210            FOR J=2 TO 7
7215               READ Cset(I,J)
7220            NEXT J
7225         NEXT I
7230      END IF
7235   !   SET THE DEFAULT CRT PLOT PARAMETERS AND PLOT A BLANK GRAPH
7240      Plot_view=0          !  FULL PLOT
7245      Plot_type=1          !  RTD TEMPERATURES
7250      Pindex=5             !  MAIN HEATER PLATE
7255      CALL Plot_prep(X1,X2,Xinc,Y1,Y2,Yinc,Xtit$,Ytit$)
7260      CALL Pblank(X1,X2,Xinc,Y1,Y2,Yinc,Xtit$,Ytit$)
7265    SUBEND
7270   ! ///////////////////////////////////////////////////////////////////////
7275    DEF FNTemp_rtd(Rtd_num,Ohms)
7280   !  THIS ROUTINE CONVERTS OHMS TO TEMPERATURE (C)
7285   !  FOR THE ROSEMOUNT PLATINUM RTD'S USING THE ITPS-68 FORM OF THE
7290   !  CALLENDAR-VAN DUSEN EQUATION.
7295      COM /Conv_dat/ R0(*),Alpha(*),Delta(*)
7300      ON ERROR GOTO Temp_rtd_err
7305      R=R0(Rtd_num)
7310      A=Alpha(Rtd_num)
7315      D=Delta(Rtd_num)
7320      A1=D/10000
7325      B1=-1-D/100
```

```
7330      C1=(Ohms-R)/(R*A)
7335      Sq=B1↑2-4*A1*C1
7340      T1=(-B1-SQR(Sq))/(2*A1)
7345      T2=T1
7350      Cnt=0
7355      REPEAT
7360         Cnt=Cnt+1
7365         T3=T2
7370         T2=T1+.045*(T2/100)*(T2/100-1)*(T2/419.58-1)*(T2/630.74-1)
7375         IF Cnt=11 THEN
7380            PRINT "  RTD ITERATIONS >10"
7385            CALL Err_record("SUB 'FNTemp_rtd' ITERATIONS>10")
7390            BEEP 83*15,.2
7395         END IF
7400      UNTIL ABS(T3-T2)<.0001 OR Cnt>10
7405      OFF ERROR
7410      RETURN T2
7415 Temp_rtd_err:    !
7420      PRINT "  TEMP_RTD ERROR ";ERRM$
7425      CALL Err_record(ERRM$)
7430      GOTO 7405
7435   FNEND
7440  ! ////////////////////////////////////////////////////////////////////////////
7445   DEF FNTemp_tc(Emf)
7450  !  THIS SUBROUTINE CONVERTS tc EMF (VOLTS) TO TEMP. (C)
7455  !  USING CONVERSION CALIBRATIONS FROM THE MEASUREMENT THERMOCOUPLE
7460  !  THERMOPILES AND NBS MONOGRAPH 161 (see 'Sys_init').
7465      COM /Sb2/ I_rtd,Tref,Emf_ref
7470      COM /Tcdat/ Tc_data(*),Tc_correction(*),Quartic_data(*)
7475      ON ERROR GOTO Temp_tc_err
7480      V=(1.0E+6)*(Emf+Emf_ref)       ! CONVERT TO MICROVOLTS
7485      SELECT V
7490      CASE <10592
7495         R2=1
7500      CASE >=10592
7505         R2=2
7510      END SELECT
7515      T=0.
7520    !        FIRST TEMP. APPROX. (K)
7525      IF V<>0 THEN
7530         FOR I=1 TO 5
7535            T=T+Quartic_data(I,R2)*(V↑(I-1))
7540         NEXT I
7545      ELSE
7550         T=Quartic_data(1,R2)
7555      END IF
7560      T=T-273.15                       ! CONVERT TO degrees Celsius
7565      Cnt=0
7570      REPEAT
7575         Micv=Tc_correction(1)
7580         Deriv=Tc_data(1)
7585         FOR I=1 TO 9
7590            Micv=Micv+Tc_data(I)*(T↑I)
7595            IF I>1 AND I<5 THEN Micv=Micv+Tc_correction(I)*(T↑(I-1))
7600            IF I>1 THEN Deriv=Deriv+I*Tc_data(I)*(T↑(I-1))
7605         NEXT I
7610         T2=T
7615         IF Deriv<>0 THEN T=T+(V-Micv)/Deriv
7620         Cnt=Cnt+1
7625         IF Cnt=11 THEN
7630            PRINT "  FNTemp_tc SUB T.C. ITERATIONS >10"
7635            CALL Err_record("SUB 'FNTemp_tc' ITERATIONS>10")
7640            BEEP 83*15,.2
7645         END IF
7650      UNTIL ABS(T2-T)<.0001 OR Cnt>10
7655      RETURN T
7660 Temp_tc_err:     !
7665      OFF ERROR
7670      PRINT " TEMP_TC SUBROUTINE ERROR : "&ERRM$
7675      CALL Err_record(ERRM$)
7680   FNEND
7685  ! ////////////////////////////////////////////////////////////////////////////
```

```
7690   SUB Ref_rtd(V,Tmr)
7695   |  THIS SUBROUTINE TAKES THE REF. BLOCK DVM VOLTAGE READING (V) CONVERTS
7700   |  THE VALUE TO TEMPERATURE, AND THEN STORES IT IN THE Rtdat ARRAY.
7705       COM /Fl/ Flag$
7710       COM /Sb1/ T0,Td0
7715       COM /Sb2/ I_rtd,Tref,Emf_ref
7720       COM /Mc1/ Ts,Ne,Edat(*),Pdat(*),Nr,Rtdat(*),Nf,Fedat(*),Ftdat(*),Kdat(*),Tme(*)
,Nrmax,Nfmax,Tlim
7725       ON ERROR GOTO Ref_rtd_err
7730       Ohms=ABS(V/I_rtd)
7735       Nr=Nr+1
7740       Rtd_num=1
7745       Tref2=FNTemp_rtd(Rtd_num,Ohms)
7750       IF ABS(Tref-Tref2)<10. OR Tref=0. THEN
7755          Tref=Tref2
7760          Emf_ref=FNEmf_tc(Tref,1)
7765       ELSE
7770          PRINT "  BAD READING FOR THE REFERENCE BLOCK TEMPERATURE !!"
7775          CALL Err_record("BAD READING FOR REFERENCE BLOCK TEMPERATURE")
7780          BEEP 83*5,.3
7785       END IF
7790       IF Nr<=Nrmax THEN
7795          Rtdat(2,Nr)=Tmr
7800          Rtdat(1,Nr)=Tref
7805       ELSE
7810          PRINT "    ISO. BLOCK STORAGE ARRAY IS FULL"
7815          CALL Err_record("ISO. BLOCK DATA STORAGE ARRAY IS FULL")
7820          BEEP 83*5,.1
7825       END IF
7830       SUBEXIT
7835 Ref_rtd_err:      !
7840       OFF ERROR
7845       PRINT " REF_RTD SUBROUTINE ERROR : "&ERRM$
7850       CALL Err_record(ERRM$)
7855   SUBEND
7860   ! ////////////////////////////////////////////////////////////////////////
7865   SUB Final_averages
7870   !       THIS SUB CALCULATES THE AVERAGED VALUES FOR PLATE TEMPERATURES,
7875   !       POWER, AND THERMAL CONDUCTIVITY.  THESE VALUES ARE THEN ASSIGNED
7880   !       TO THEIR CORRESPONDING DISK STORAGE ARRAY ELEMENTS.
7885       COM /Dt1/ File_specs$(*),Made$(*),Gas$(*)
7890       COM /Mc1/ Ts,Ne,Edat(*),Pdat(*),Nr,Rtdat(*),Nf,Fedat(*),Ftdat(*),Kdat(*),Tme(*)
,Nrmax,Nfmax,Tlim
7895       COM /Mc6/ Ntm,Ntr,Ntp,Ntz
7900       COM /Sb1/ T0,Td0
7905       COM /Sb2/ I_rtd,Tref,Emf_ref
7910       COM /Sb3/ Fd(*),Tavg_scan
7915       IF Nf<1 THEN SUBEXIT
7920       ON ERROR GOTO Calc_err
7925       ALLOCATE Tlims(1:2)
7930       Tcalc=TIMEDATE     ! RECORD WHEN THE SUB WAS ENTERED (TO TIME EXIT)
7935       WHILE Tme(Nf)=0
7940          Nf=Nf-1
7945       END WHILE
7950       Tlast=Tme(Nf)
7955       Tend=1         ! INDEX FOR TLIMS(*) , BEGINNING & END OF AVG. SECTION
7960       Tstep=1        ! TIME TOGGLE SIZE (MIN.)
7965       Tlims(2)=Tme(Nf)
7970       Tlims(1)=Tlims(2)-Tavg_scan
7975       IF Tlims(1)<0 THEN Tlims(1)=0
7980       IF Tlims(2)-Tlims(1)<=0 THEN SUBEXIT
7985       PRINT USING "@,10/"
7990       BEEP 83*11,.3
7995       DISP " SELECT THE TIME ENDPOINTS FOR THE FINAL DATA AVERAGING INTERVAL (BEFORE
T-0) "
8000   !          REDEFINE SOFTKEYS
8005       FOR I=0 TO 9
8010          ON KEY I LABEL "" GOSUB 8970
8015       NEXT I
8020       ON KEY 9 LABEL "PERFORM AVGS. " GOTO Take_avg
8025       ON KEY 1 LABEL " CHANGE PLOT  " GOTO 8055
8030       ON KEY 0 LABEL " SELECT TLIM. " GOTO 8070
```

```
8035      ON KEY 2 LABEL " SELECT T STEP" GOTO 8100
8040      ON KEY 3 LABEL "  TOGGLE UP   " GOTO 8115
8045      ON KEY 8 LABEL "  TOGGLE DOWN " GOTO 8140
8050      GOTO 8160
8055      CALL Plot_switch
8060      Tcalc=TIMEDATE
8065      GOTO 8160
8070      IF Tend=1 THEN            !  SELECT Tend (1=BEG. OR 2=END OF THE RANGE)
8075         Tend=2
8080      ELSE
8085         Tend=1
8090      END IF
8095      GOTO 8160
8100      Tstep=Tstep+1             !  SELECT Tstep
8105      IF Tstep>10 THEN Tstep=1
8110      GOTO 8160
8115      Tlims(Tend)=Tlims(Tend)+Tstep*60
8120  !          DON'T LET THE TEMP. SPAN FALL BELOW TAVG_SCAN SECONDS
8125      IF Tlims(2)-Tlims(1)<Tavg_scan THEN Tlims(Tend)=Tlims(Tend)-Tstep*60
8130      IF Tlims(2)>Tlast THEN Tlims(2)=Tlast
8135      GOTO 8160
8140      Tlims(Tend)=Tlims(Tend)-Tstep*60
8145  !          DON'T LET THE TEMP. SPAN FALL BELOW TAVG_SCAN SECONDS
8150      IF Tlims(Tend)<Td0 THEN Tlims(Tend)=Tlims(Tend)+Tstep*60
8155      IF Tlims(2)-Tlims(1)<Tavg_scan THEN Tlims(Tend)=Tlims(Tend)+Tstep*60
8160      IF Tend=1 THEN
8165         Te$="LOWER"
8170      ELSE
8175         Te$="UPPER"
8180      END IF
8185      ON KEY 5 LABEL "    "&Te$ GOSUB 8970
8190      ON KEY 6 LABEL "  "&VAL$(INT(Tlims(Tend)/60))&" min." GOSUB 8970
8195      ON KEY 7 LABEL "  "&VAL$(Tstep)&" min" GOSUB 8970
8200      LOOP
8205         ON KEY 4 LABEL "  T - "&VAL$(300-INT(TIMEDATE-Tcalc)) GOSUB 8970
8210      EXIT IF TIMEDATE-Tcalc>300
8215      END LOOP
8220 !-------------------------------
8225 Take_avg:    !        PERFORM AVG. CALC.
8230   !        FIND THE END ARRAY INDEXES
8235      GRAPHICS OFF
8240      DISP "     DATA AVERAGING NOW BEING PERFORMED "
8245      WAIT .2
8250      Fd(28)=Tlims(2)-Tlims(1)  !    AVERAGING INTERVAL (SEC.)
8255      Fd(29)=Tlims(1)           !    START POINT FOR DATA AVERAGING INTERVAL
8260      Nstart=1
8265      WHILE (Tlims(1)>Tme(Nstart)+30) AND Nstart<Nf
8270         Nstart=Nstart+1
8275      END WHILE
8280      Nfinish=1
8285      WHILE (Tlims(2)>Tme(Nfinish)-30) AND Nfinish<Nf
8290         Nfinish=Nfinish+1
8295      END WHILE
8300  !     AVERAGE THE TEMP'S FROM THE tc's   (WITH THE deltT CORRECTION)
8305      Nsize=Nfinish-Nstart+1
8310      ALLOCATE Temp(1:Nsize),Tsec(1:Nsize),Slope(1:5),Avg(1:5),Sd(1:5),Avgcorr(1:4),S
pmt$[20]
8315      FOR I=1 TO 4
8320         Corrsum=0
8325         FOR J=Nstart TO Nfinish
8330            Temp(J-Nstart+1)=Ftdat(I*2-1,J)+Ftdat(I*2,J)
8335            Tsec(J-Nstart+1)=Tme(J)
8340            Corrsum=Corrsum+Ftdat(I*2,J)
8345         NEXT J
8350         Avgcorr(I)=Corrsum/Nsize
8355         CALL Linear(Tsec(*),Temp(*),Nsize,D1,Slpe,Mean,Stddev)
8360         Avg(I)=Mean     !    I=1=UAP, 2=UMP, 3=LMP, 4=LAP
8365         Slope(I)=Slpe
8370         Sd(I)=Stddev
8375         Fd(15-I)=Mean
8380      NEXT I
8385  !      COMPUTE POWER AVERAGE
```

```
8390       Nst=INT(Tlims(1)/Ts)
8395       IF Nst<1 THEN Nst=1
8400       Nfin=INT(Tlims(2)/Ts)
8405       IF Nfin>Ne THEN Nfin=Ne
8410       Npwr=Nfin-Nst+1
8415       ALLOCATE Pwr(1:Npwr),Ptm(1:Npwr)
8420       FOR I=Nst TO Nfin
8425          Pwr(I-Nst+1)=Pdat(1,I)*Pdat(2,I)
8430          Ptm(I-Nst+1)=I*Ts
8435       NEXT I
8440       CALL Linear(Ptm(*),Pwr(*),Npwr,D1,Slpe,Mean,Stddev)
8445       Avg(5)=Mean
8450       Slope(5)=Slpe
8455       Sd(5)=Stddev
8460       Fd(9)=Mean
8465       Q=Fd(9)*1000.       !  POWER CONVERTED TO MILLIWATTS FOR THE SUB 'K_ghp'
8470       T1=Fd(11)           !  LAP TEMP.
8475       T2=Fd(12)           !  LMP TEMP.
8480       T3=Fd(13)           !  UMP TEMP.
8485       T4=Fd(14)           !  UAP TEMP.
8490       Dia=Fd(8)*100.
8495       Dx=Fd(3)*100.
8500       Dr=Fd(21)*100.
8505       Sc=Fd(22)           !  PLATE SPACER CODE
8510       Rc=Fd(15)           ! RUN MODE (DBLE SIDED, TOP, BOT.)
8515       CALL K_ghp(K,Tlo,Thi,Dxc,Acor,T1,T2,T3,T4,Q,Dia,Dx,Dr,Sc,Rc,0)
8520       Fd(1)=(Thi+Tlo)/2     ! deg C
8525       Fd(2)=K/1000.         ! THERMAL CONDUCTIVITY STORED IN W/(m*K)
8530       Fd(16)=Thi            ! deg C
8535       Fd(17)=Tlo            ! deg C
8540       Fd(23)=Dxc/100.       !  CORRECTED SAMPLE THICKNESS Dx STORED IN meters
8545       Fd(25)=Acor/10000.    !  CORRECTED MAIN PLATE AREA (m↑2)
8550    !-------------------------------
8555    !       CALC STD.DEV. OF K
8560       Delt_frac=(SQR(Sd(1)↑2+Sd(2)↑2+Sd(3)↑2+Sd(4)↑2))/(T2+T3-T1-T4)
8565       Q_frac=Sd(5)/Fd(9)
8570       K_frac=SQR(Delt_frac↑2+Q_frac↑2)
8575       Kstd_dev=K_frac*Fd(2)*1000.   !    mW/m*K
8580    !-------------------------------
8585    !          PRINT OUT K CALC RESULTS
8590       PRINTER IS 701
8595       PRINT USING "@,2/"
8600       PRINT "    FILE NAME : ";File_specs$(1)[1,7]
8605       PRINT
8610       PRINT "    ";File_specs$(1)[11,80]
8615       PRINT "    ";File_specs$(3)
8620       PRINT "    ";File_specs$(5)
8625       PRINT
8630       PRINT "             GENERAL FILE SPECIFICATIONS:"
8635       PRINT "    SAMPLE THICKNESS (cm)    ; UNCORRECTED=";Fd(3)*100.;",   CORRECTED="
;INT(Fd(23)*100000.)/1000.
8640       Area_unc=INT((PI*Fd(8)↑2/4)*10000.*1000.)/1000.
8645       Area_cor=INT(Fd(25)*10000.*1000.)/1000.
8650       PRINT "    MAIN PLATE AREA (cm↑2) ; UNCORRECTED=";Area_unc;",   CORRECTED=";Are
a_cor
8655       PRINT "    AREA DENSITY (kg/m↑2)=";Fd(4);",  CORRECTED BULK DENSITY (kg/m↑3)=";
INT(Fd(4)/Fd(23)*1000.)/1000.
8660       IF Fd(22)=1 THEN
8665          Spmt$="QUARTZ"
8670       ELSE
8675          Spmt$="STAINLESS STEEL"
8680       END IF
8685       PRINT "    PLATE SPACER MATERIAL : ";Spmt$
8690       PRINT "    FILL GAS : ";File_specs$(4)[1;20];" ,   PRESSURE (mmHg) = ";Fd(5)
8695       PRINT "    SPECIMEN AND SPECIMEN GUARD CODES :"
8700       PRINT "          TOP SPECIMEN CODE    : ";File_specs$(4)[21,30]
8705       PRINT "          TOP GUARD CODE       : ";File_specs$(4)[41,50]
8710       PRINT "          BOTTOM SPECIMEN CODE : ";File_specs$(4)[31,40]
8715       PRINT "          BOTTOM GUARD CODE    : ";File_specs$(4)[51,60]
8720       PRINT
8725       PRINT "    CORRECTED AVERAGE PLATE TEMPERATURES (degC)  AND TEMP. CORRECTIONS "
8730       PRINT
```

```
8735       PRINT "       UPPER AUX. PLATE TEMP.=";INT(Avg(1)*10000.)/10000.;", STD.DEV.=";
INT(Sd(1)*10000.)/10000.
8740       PRINT "           TEMP. CORRECTION=";INT(Avgcorr(1)*10000.)/10000.;",  dT/dt(de
g/hr)=";INT(3600.*Slope(1)*10000.)/10000.
8745       PRINT "       UPPER MAIN PLATE TEMP.=";INT(Avg(2)*10000.)/10000.;", STD.DEV.=";
INT(Sd(2)*10000.)/10000.
8750       PRINT "           TEMP. CORRECTION=";INT(Avgcorr(2)*10000.)/10000.;",  dT/dt(de
g/hr)=";INT(3600.*Slope(2)*10000.)/10000.
8755       PRINT "       LOWER MAIN PLATE TEMP.=";INT(Avg(3)*10000.)/10000.;", STD.DEV.=";
INT(Sd(3)*10000.)/10000.
8760       PRINT "           TEMP. CORRECTION=";INT(Avgcorr(3)*10000.)/10000.;",  dT/dt(de
g/hr)=";INT(3600.*Slope(3)*10000.)/10000.
8765       PRINT "       LOWER AUX. PLATE TEMP.=";INT(Avg(4)*10000.)/10000.;", STD.DEV.=";
INT(Sd(4)*10000.)/10000.
8770       PRINT "           TEMP. CORRECTION=";INT(Avgcorr(4)*10000.)/10000.;",  dT/dt(de
g/hr)=";INT(3600.*Slope(4)*10000.)/10000.
8775       Sdmt=(SQR(Sd(1)↑2+Sd(2)↑2))/SQR(Nsize)
8780       PRINT "       UPPER delta T=";INT((Avg(2)-Avg(1))*10000.)/10000.;", STD. DEV. O
F MEAN=";INT(Sdmt*10000.)/10000.
8785       Sdmb=(SQR(Sd(3)↑2+Sd(4)↑2))/SQR(Nsize)
8790       PRINT "       LOWER delta T=";INT((Avg(3)-Avg(4))*10000.)/10000.;", STD. DEV. O
F MEAN=";INT(Sdmb*10000.)/10000.
8795       Sdmtot=SQR(Sdmt↑2+Sdmb↑2)
8800       Psdmtot=100.*SQR(Sdmt↑2+Sdmb↑2)/(T2-T1+T3-T4)
8805       Sddelt=INT(Delt_frac*(T2-T1+T3-T4)*10000.)/10000.
8810       Psddelt=INT(Delt_frac*100.*100.)/100.
8815       PRINT "       TOTAL delta T (T2-T1+T3-T4)=";INT((T2-T1+T3-T4)*10000.)/10000.
8820       PRINT "          STD. DEV. OF DelT =";Sddelt;",    % STD.DEV. OF DelT =";Psddel
t
8825       PRINT "          STD. DEV. OF THE MEAN=";INT(Sdmtot*10000.)/10000.;",   % STD.D
EV. OF THE MEAN =";INT(Psdmtot*10000.)/10000.
8830       PRINT
8835       PRINT "    AVG. MAIN HEATER PLATE POWER (mW)=";INT(1.E+6*Avg(5))/1000.
8840       PRINT "           STD.DEV. OF Q =";INT(1.E+6*Sd(5))/1000.;",    % STD.DEV. OF Q
=";INT(Q_frac*10000.)/100.
8845       Sdqmean=INT((1.E+6*Sd(5))/SQR(Npwr))/1000.
8850       Psdqmean=INT((1.E+8*Sd(5))/SQR(Npwr)/(1.E+3*Avg(5)))/1000.
8855       PRINT "           STD.DEV. OF THE MEAN =";Sdqmean;",    % STD.DEV. OF THE MEAN =
";Psdqmean
8860       PRINT "           dP/dt(mW/hr)=";INT(1.E+6*3600.*Slope(5))/1000.
8865       PRINT "    DATA AVERAGING INTERVAL : ";INT(Fd(28)*100./60.)/100.;" MINUTES"
8870       PRINT "    START POINT OF INTERVAL : ";INT(Fd(29)*100./60.)/100.;" MINUTES "
8875       PRINT
8880       PRINT "    FINAL HEATER RESISTANCE : ";INT(Fd(10)*10000.)/10000.;" OHMS"
8885       PRINT
8890       PRINT "    IG THERMOPILE FINAL READINGS :"
8895       PRINT "             UPPER PILE : ";INT(Fd(20)*1.E+9)/1000.;" microvolts"
8900       PRINT "             LOWER PILE : ";INT(Fd(18)*1.E+9)/1000.;" microvolts"
8905       PRINT "             TOTAL PILE : ";INT(Fd(19)*1.E+9)/1000.;" microvolts"
8910       PRINT
8915       Dtavg=INT((Fd(16)-Fd(17))*10000.)/10000.
8920       PRINT "    Thi (degC) = ";INT(Fd(16)*10000.)/10000.;" ,  Tlo (degC) = ";INT(Fd(
17)*10000.)/10000.;" ,  delT=";Dtavg
8925       IF Fd(2)<10 AND Fd(2)>-1 THEN
8930         PRINT USING "23A,4D.DDD,25A,4D.DD,8A";"    AVG. TEMP.(C) IS :",Fd(1)," ,  TH
ERMAL COND. (k) = ",Fd(2)*1000.," mW/(m*K)"
8935       ELSE
8940         PRINT USING "23A,4D.DDD,40A";"    AVG. TEMP.(C) IS :",Fd(1)," ,  THERMAL CON
D. (k) = 9999.99 mW/(m*K)"
8945       END IF
8950       PRINT "           STD.DEV. OF k =";INT(Kstd_dev*1000.)/1000.;",   % STD.DEV. OF
 k =";INT(K_frac*100.*100.)/100.
8955       PRINTER IS 1
8960       OFF ERROR
8965       SUBEXIT
8970       Tcalc=TIMEDATE
8975       RETURN
8980 Calc_err:    !
8985       PRINTER IS 1
8990       PRINT "SUB 'FINAL AVERAGES' ERROR : "&ERRM$
8995       CALL Err_record(ERRM$)
9000       BEEP 83*5,.3
```

```
9005      WAIT 2
9010   SUBEND
9015  ! ///////////////////////////////////////////////////////////////////////////
9020   DEF FNEmf_tc(Temp,OPTIONAL Ref_flag)
9025  !  THIS SUBROUTINE TEMP.(C) TO EMF.(VOLTS)
9030  !  USING CONVERSION CALIBRATIONS FROM THE MEASUREMENT THERMOCOUPLE
9035  !  THERMOPILES AND NBS MONOGRAPH 161 (see 'Sys_init').
9040  !         IF A SECOND PARAMETER IS GIVEN TO THE SUB (ie. Ref_flag) THEN
9045  !         THE REF. EMF. IS NOT SUBTRACTED FROM THE CALC. EMF
9050      COM /Sb2/ I_rtd,Tref,Emf_ref
9055      COM /Tcdat/ Tc_data(*),Tc_correction(*),Quartic_data(*)
9060      ON ERROR GOTO Emf_tc_err
9065      Emf=0.
9070      IF Temp<>0 THEN
9075         FOR I=1 TO 9
9080            Emf=Emf+Tc_data(I)*(Temp↑I)
9085            IF I<5 THEN Emf=Emf+Tc_correction(I)*(Temp↑(I-1))
9090         NEXT I
9095      ELSE
9100         Emf=Tc_correction(1)
9105      END IF
9110      IF NPAR=1 THEN Emf=Emf-Emf_ref*1.0E+6    ! SUBTRACT REF. EMF
9115      Emf=Emf/(1.0E+6)        !  CONVERT TO VOLTS
9120      RETURN Emf
9125 Emf_tc_err:      !
9130      OFF ERROR
9135      PRINT " EMF_TC SUBROUTINE ERROR : "&ERRM$
9140      CALL Err_record(ERRM$)
9145   FNEND
9150  ! ///////////////////////////////////////////////////////////////////////////
9155   SUB Record_data
9160  !  THIS SUBROUTINE STORES TEMPERATURE AND EMF. DATA FOR A RUN ON
9165  !  A FLOPPY DISK FILE.
9170      COM /Mc1/ Ts,Ne,Edat(*),Pdat(*),Nr,Rtdat(*),Nf,Fedat(*),Ftdat(*),Kdat(*),Tme(*)
,Nrmax,Nfmax,Tlim
9175      COM /Dt1/ File_specs$(*),Mode$(*),Gas$(*)
9180      COM /Mc5/ Op_mode(*),Htemp(*),Ltemp(*),Ht_mode(*),File_num(*),Set_temp(*)
9185      COM /Sb1/ T0,Td0
9190      COM /Sb2/ I_rtd,Tref,Emf_ref
9195      COM /Sb3/ Fd(*),Tavg_scan
9200      COM /Rn/ Run
9205      Drive_change=0
9210      MASS STORAGE IS ":INTERNAL,4,0"
9215      ON ERROR GOSUB Record_err
9220      Filename$=File_specs$(1)[1;7]
9225      Fl_orig$=Filename$
9230      File_specs$(2)[1]="HIGH TEMP.(C)="&VAL$(Htemp(Run))&" ; LOW TEMP.(C)="&VAL$(Lte
mp(Run))&" ; MAIN HEATER CONTROL : "
9235      IF Ht_mode(Run)=1 THEN
9240         File_specs$(2)[1]=File_specs$(2)&"CONST. TEMP."
9245      ELSE
9250         File_specs$(2)[1]=File_specs$(2)&"CONST. POWER"
9255      END IF
9260  !  TRIM ARRAYS TO REQUIRED SIZE
9265      IF Nr=0 THEN Nr=1
9270      IF Nf=0 THEN
9275         Nf=1
9280         Nstart=1
9285         Npts=Ne
9290      ELSE
9295         Nstart=INT(Td0/Ts)
9300         Npts=Ne-Nstart+1
9305      END IF
9310      Fd(27)=Td0                          !  TIME OF DATA RUN START
9315   !      STORE THE POWER DATA FROM TIME Td0
9320      ALLOCATE P(1:2,1:Npts)
9325      FOR I=1 TO Npts
9330         FOR J=1 TO 2
9335            P(J,I)=Pdat(J,I-1+Nstart)
9340         NEXT J
9345      NEXT I
9350      ALLOCATE Tm(1:Nf),Ft(1:4,1:Nf),Fe(1:8,1:Nf),Rt(1:2,1:Nr)
```

```
9355      FOR I=1 TO Nf
9360         FOR J=1 TO 8                        !  STORE CORRECTED T's AND EMF's
9365            IF J<5 THEN Ft(J,I)=Ftdat(J*2-1,I)+Ftdat(J*2,I)
9370            Fe(J,I)=Fedat(J,I)
9375         NEXT J
9380         Tm(I)=Tme(I)
9385      NEXT I
9390      FOR I=1 TO Nr
9395         FOR J=1 TO 2
9400            Rt(J,I)=Rtdat(J,I)
9405         NEXT J
9410      NEXT I
9415  !  OPEN A FILE ON THE DISK
9420      Nrec=INT((8.*(2*Npts+2*Nr+8*Nf+Nf+4*Nf+4+30)+5*84+4)/256.+1)
9425      CREATE BDAT Filename$,Nrec
9430      ASSIGN @Io_path TO Filename$
9435      PRINT "  DATA IS BEING STORED ON THE DISK FILE NAME : ";Filename$
9440      PRINT USING "3/"
9445  !  STORE THE DATA ON THE DISK FILE
9450      OUTPUT @Io_path;File_specs$(*),Fd(*),Nf,Tm(*),Ft(*),Fe(*),Nr,Rt(*),Npts,P(*)
9455      ASSIGN @Io_path TO *
9460      IF Fl_orig$<>Filename$ THEN
9465         PRINTER IS 701
9470         PRINT USING "2/"
9475         PRINT "         ****************************************************"
9480         PRINT "         ****      THE FILE NAME HAS BEEN CHANGED!!      ****"
9485         PRINT "         ****       ( DUE TO FILENAME DUPLICATION )      ****"
9490         PRINT "         ****                                            ****"
9495         PRINT "         ****      THE OLD FILENAME WAS : ";Fl_orig$;"       ****"
9500         PRINT "         ****                                            ****"
9505         PRINT "         ****      THE NEW FILENAME IS : ";Filename$;"        ****"
9510         PRINT "         ****************************************************"
9515         PRINT
9520         PRINTER IS 1
9525      END IF
9530      DISP "  DATA STORAGE HAS BEEN COMPLETED."
9535      OFF ERROR
9540      WAIT 3
9545      DISP " "
9550      MASS STORAGE IS ":INTERNAL,4,0"
9555      SUBEXIT
9560 Record_err: !
9565      BEEP 81.38*16,1
9570      T_delay=TIMEDATE
9575   ! SET THE COMPUTER SELF ACTION (CSA) TIME DELAY VARIABLE (IN SEC.)
9580      Delay_int=180
9585   ! SET THE CSA TIME DELAY FLAG (1=DELAY ON, 0=DELAY DISABLE)
9590      Do_it=1
9595      ON ERROR GOSUB Record_err
9600      SELECT ERRN
9605      CASE 54
9610         PRINT "DUPLICATE FILE NAME (";Filename$;")"
9615         PRINT "       HIT 'CONTINUE'"
9620         PRINT " THEN INPUT A NEW FILE NUMBER."
9625         PRINT USING "10/"
9630         ON KBD GOTO 9660
9635         BEEP 81.38*20,.8
9640         WAIT 1
9645         IF KBD$<>"" THEN GOTO 9660
9650         IF Do_it AND TIMEDATE-T_delay>Delay_int THEN GOTO Take_action
9655         GOTO 9635
9660         OFF KBD
9665         INPUT " ENTER THE NEW FILE NUMBER : ",Fl_num
9670         Res$=VAL$(Fl_num)
9675         File_specs$(1)[5,7]="000"
9680         File_specs$(1)[8-LEN(Res$),7]=Res$
9685         Filename$=File_specs$(1)[1,7]
9690         GOTO 10100
9695      CASE 64
9700         PRINT "DISK IN DRIVE :INTERNAL,4,";Drive_change;" IS FULL ... REPLACE DISK!"
9705         IF Drive_change=1 THEN
9710            MASS STORAGE IS ":INTERNAL,4,0"
```

```
9715            Do_it=0
9720            PRINT
9725            PRINT "THE DISKS IN BOTH DISK DRIVES ARE FULL!!"
9730            PRINT "THE MASS STORAGE DEVICE HAS BEEN RE-ASSIGNED TO THE RIGHT"
9735            PRINT "DISK DRIVE, SO REPLACE THE ':INTERNAL,4,0' DISK!!"
9740            PRINT
9745         ELSE
9750            GOTO Take_action
9755         END IF
9760      CASE 80
9765         PRINT "NO DISK PRESENT OR DRIVE NOT CLOSED!"
9770         IF Drive_change=1 THEN
9775            Do_it=0
9780            MASS STORAGE IS ":INTERNAL,4,0"
9785            PRINT
9790            PRINT "PLACE A DISK IN THE RIGHT DRIVE (OR CLOSE THE DOOR)!!"
9795            PRINT
9800         END IF
9805      CASE 85
9810         PRINT "DISK IS NOT INITIALIZED!!"
9815         PRINT "REPLACE IT WITH ONE THAT HAS BEEN INITIALIZED."
9820         PRINT "          (OR INITIALIZE IT)"
9825      CASE ELSE
9830         PRINTER IS 1
9835         PRINT "YOU HAVE ENCOUNTERED ERROR NUMBER ";ERRN
9840         PRINT "   ";ERRM$
9845         Do_it=0
9850      END SELECT
9855      DISP " HIT ANY KEY TO TERMINATE THIS INFERNAL BEEPING!!"
9860      Sys_on=1
9865      ON KBD GOTO Rezume2
9870      FOR I=5 TO 50 STEP 5
9875         BEEP 81.38*I,.01
9880      NEXT I
9885      FOR I=45 TO 20 STEP -5
9890         BEEP 81.38*I,.01
9895      NEXT I
9900      BEEP 81.38*25,.5
9905      Tbeep=TIMEDATE
9910      IF TIMEDATE-Tbeep<.6 THEN GOTO 9910
9915      IF KBD$<>"" THEN GOTO Rezume2
9920      IF Do_it AND TIMEDATE-T_delay>Delay_int THEN GOTO Take_action
9925      IF Do_it=0 AND TIMEDATE-T_delay>Delay_int AND Sys_on THEN
9930         CALL Sys_shutdown
9935         Sys_on=0
9940      END IF
9945      GOTO 9870
9950 Rezume2:     !
9955      OFF KBD
9960      DISP " "
9965      PRINT
9970      PRINT "                    CORRECT THE ERROR, IF POSSIBLE ...."
9975      PRINT "                            THEN HIT 'CONTINUE'"
9980      PRINT
9985      PRINT "         (IF THE ERROR IS IRRECOVERABLE ...STILL HIT 'CONTINUE')"
9990      PRINT
9995      PAUSE
10000     INPUT " WAS THE ERROR RECOVERABLE (Y/N)?",Res$
10005     IF Res$="N" THEN
10010        Flag$="RECORD_DATA"
10015        SUBEXIT
10020     END IF
10025     GOTO 10100
10030 Take_action:   !
10035     OFF KBD
10040     DISP " "
10045     IF ERRN=54 THEN
10050        Fl_num=VAL(File_specs$(1)[5,7])
10055        IF Fl_num<999 THEN Fl_num=Fl_num+1
10060        File_specs$(1)[8-LEN(VAL$(Fl_num)),7]=VAL$(Fl_num)
10065        Filename$=File_specs$(1)[1,7]
10070     END IF
```

```
10075      IF ERRN=64 OR ERRN=80 THEN
10080         Drive_change=1
10085         MASS STORAGE IS ":INTERNAL,4,1"
10090      END IF
10095      IF ERRN=85 THEN INITIALIZE ":INTERNAL"
10100      RETURN
10105  SUBEND
10110 ! ////////////////////////////////////////////////////////////////////////
10115  SUB Pack_queue(Q(*),Nq,Qseq(*),Nqs)
10120 !     THIS SUB LOADS THE QUEUE Q(*) WITH THE ARRAY QSEQ(*), ADDING IT
10125 !     AFTER ANY ENTRIES ALREADY PRESENT IN Q(*). AN ENTRY IN Q(*) THAT
10130 !     IS LESS THAN 0 IS CONSIDERED EMPTY (SINCE JULIE RELAY CHANNELS
10135 !     ARE NUMBERED 0 TO 19).
10140      ON ERROR GOTO Pack_err
10145      N=0
10150      REPEAT
10155         N=N+1
10160      UNTIL (N=Nq) OR (Q(N)<0)
10165      IF Nq-N<Nqs THEN
10170         DISP "CHANNEL QUEUE OVERFLOW!"
10175      END IF
10180      IF N=Nq AND Q(N)>=0 THEN SUBEXIT
10185      Full=N+Nqs-1
10190      IF Full>Nq THEN Full=Nq
10195      FOR I=N TO Full
10200         Q(I)=Qseq(I-N+1)
10205      NEXT I
10210      SUBEXIT
10215 Pack_err:     !
10220      OFF ERROR
10225      PRINT " PACK_QUEUE SUBROUTINE ERROR : "&ERRM$
10230      CALL Err_record(ERRM$)
10235  SUBEND
10240 ! ////////////////////////////////////////////////////////////////////////
10245  SUB Poweron
10250 !        THIS SUB TURNS 'ON' THE POWER FLAG
10255      COM /Manual/ Powerflag,Vreading(*)
10260      Powerflag=1
10265  SUBEND
10270 ! ////////////////////////////////////////////////////////////////////////
10275  SUB Poweroff
10280 !        THIS SUB TURNS 'OFF' THE POWER FLAG
10285      COM /Fl/ Flag$
10290      COM /Manual/ Powerflag,Vreading(*)
10295      ON ERROR GOTO Poweroff_err
10300      FOR I=7 TO 11
10305         Write_io(723,"OP,"&VAL$(I)&",0T")
10310         WAIT .1
10315      NEXT I
10320      CALL Outseven(0)
10325      Powerflag=0
10330      SUBEXIT
10335 Poweroff_err:     !
10340      OFF ERROR
10345      Flag$="ERROR IN SUB 'POWEROFF' : "&ERRM$
10350  SUBEND
10355 ! ////////////////////////////////////////////////////////////////////////
10360  SUB Adjust
10365  !        THIS SUB IS CALLED EVERY CONTROL CYCLE BY THE 'ON CYCLE' COMMAND.
10370  !        IT READS THE TEMP. OF ALL THE CONTROL LOOPS AND SETS THE POWER
10375  !        SUPPLY LEVELS.
10380      COM /Ioscan/ Queue(*),Nq,Qseq1(*),Nqs1,Qseq2(*),Nqs2,Qseq3(*),Nqs3,Qseq4(*),Nqs
4,Qseq5(*),Nqs5,Qseq6(*),Nqs6
10385      COM /Ctr1/ Cdata(*),Cset(*),Cntrl_vlim(*),Loop_label$(*),Cstr$(*),Mhvmax
10390      COM /Sdisp/ Screen_prnt
10395      COM /Manual/ Powerflag,Vreading(*)
10400      COM /Adjloc2/ Last_reading(*)
10405      COM /Constpwr/ Esum,Tsum,Tsum2,Etsum,P2n,Ntp2
10410      COM /Fl/ Flag$
10415      COM /Flgs/ Igflag,Prev_ne
10420      COM /Htr1/ Htr_res
10425      COM /Htr2/ New_htrcur
```

```
10430     COM /Instr/ Mh181,Jrl181,Mh195,Tap195,Bap195,Ig195
10435     COM /Mc1/ Ts,Ne,Edat(*),Pdat(*),Nr,Rtdat(*),Nf,Fedat(*),Ftdat(*),Kdat(*),Tme(*)
,Nrmax,Nfmax,Tlim
10440     COM /Mc2/ Heater_sres
10445     COM /Mc5/ Op_mode(*),Htemp(*),Ltemp(*),Ht_mode(*),File_num(*),Set_temp(*)
10450     COM /Mc6/ Ntm,Ntr,Ntp,Ntz
10455     COM /Rn/ Run
10460     COM /Sb1/ T0,Td0
10465     COM /Sb2/ I_rtd,Tref,Emf_ref
10470     COM /Sb3/ Fd(*),Tavg_scan
10475     DIM Power_prob(1:5)
10480     ON KBD GOSUB Nothing
10485     ON KEY 0 LABEL "" GOSUB Nothing
10490     ON KEY 5 LABEL "" GOSUB Nothing
10495     ON ERROR GOTO Adj_error
10500     Tadj=TIMEDATE
10505     Ne=Ne+1
10510     IF Ne>1 THEN Ov=Pdat(2,Ne-1)
10515     CALL Read_io(Vpwr,Tread,Mh195,"R0X",Ov)
10520     Pdat(2,Ne)=Vpwr
10525     IF New_htrcur THEN
10530        New_htrcur=0
10535        IF Pdat(1,Ne-1)<>0 THEN Fd(10)=Vpwr/Pdat(1,Ne-1)
10540        IF ABS(Fd(10)-Htr_res)>.5*Htr_res THEN
10545           CALL Err_record(" HEATER RESISTANCE VARIATION TOO GREAT : OLD="&VAL$(Htr_
res)&" , NEW (IGNORED) = "&VAL$(Fd(10)))
10550           PRINT " HEATER RESISTANCE VARIATION TOO GREAT : OLD="&VAL$(Htr_res)&" , N
EW (IGNORED) = "&VAL$(Fd(10))
10555           Fd(10)=Htr_res
10560           BEEP 83*28,.1
10565        END IF
10570        IF Screen_prnt THEN PRINT "    HEATER RESISTANCE CALC. = ";Fd(10);" OHMS"
10575        Htr_res=Fd(10)
10580     END IF
10585     Pdat(1,Ne)=Vpwr/Htr_res      ! FOR EACH PAIR OF IDENTICAL CURRENTS (Ne
10590                                  ! & Ne-1) A TRUE POWER CAN BE CALC. AT Ne.
10595     MAT Power_prob= (0.)
10600     FOR I=1 TO 5
10605        Vin=Vreading(I)
10610        IF I=1 THEN CALL Read_io(Vout,T,Ig195,"R0X",Vin)
10615        IF I=2 THEN CALL Atod_io(Vout,T,Vin)    !  A/D CARD OUTPUT (FOR OG)
10620        IF I=3 THEN CALL Read_io(Vout,T,Tap195,"R0X",Vin)
10625        IF I=4 THEN CALL Read_io(Vout,T,Bap195,"R0X",Vin)
10630        IF I=5 THEN CALL Read_io(Vout,T,Mh181,"R3B1X",Vin)
10635        Vreading(I)=Vout
10640        IF (I>1) AND (Vreading(I)<(100.*.0001)) THEN Power_prob(I)=1
10645     NEXT I
10650     IF SUM(Power_prob)=4 THEN Flag$="RTD POWER SUPPLY MALFUNCTION"
10655     IF Powerflag=1 THEN          !  ADJUST HEATERS
10660        Pid(1,Vreading(1),Ot)
10665        FOR I=2 TO 5
10670           Pid(I,Vreading(I)/I_rtd,Ot)
10675        NEXT I
10680        IF ((Ne-P2n) MOD Ntp2=0) AND (Td0<>0) AND (Ht_mode(Run)=2) THEN
10685           P2n=Ne
10690           Etslope=(Etsum*Ntp2-Esum*Tsum)/(Tsum2*Ntp2-Tsum↑2)  !  K/SEC
10695           Esum=0
10700           Tsum=0
10705           Tsum2=0
10710           Etsum=0
10715           Slope_cor=-(110*(.8+.006*Edat(5,Ne))↑.5)*Etslope/Cdata(5,2,2)
10720           IF ABS(Slope_cor/Cdata(5,2,2))>.05 THEN
10725              PRINT " SLOPE CORRECTION OVER 5%; CHANGE RESTRICTED TO 5%"
10730              Slope_cor=SGN(Slope_cor)*.05*Cdata(5,2,2)
10735           END IF
10740           Cdata(5,2,2)=Cdata(5,2,2)+Slope_cor
10745        END IF
10750        IF (Ht_mode(Run)=2) AND (Td0<>0) THEN Cdata(5,2,1)=Cdata(5,2,2)
10755        CALL Write_io(723,"OP,"&VAL$(11)&","&VAL$(Cdata(1,2,1))&"T")
10760        CALL Write_io(723,"OP,"&VAL$(8)&","&VAL$(Cdata(2,2,1))&"T")
10765        CALL Write_io(723,"OP,"&VAL$(9)&","&VAL$(Cdata(3,2,1))&"T")
10770        CALL Write_io(723,"OP,"&VAL$(10)&","&VAL$(Cdata(4,2,1))&"T")
```

```
10775       Outseven(Cdata(5,2,1))       !   MAIN PLATE
10780     END IF
10785 !        STORE RTD TEMPERATURE DATA
10790     FOR I=2 TO 5
10795       Edat(I,Ne)=FNTemp_rtd(I,Vreading(I)/I_rtd)
10800       IF Edat(I,Ne)-Set_temp(I)>100 OR Edat(I,Ne)>525 THEN Flag$="  PLATE TEMPERAT
URE OUT OF CONTROL ON CONTROL LOOP NUMBER "&VAL$(I)
10805     NEXT I
10810     IF (Td0<>0) AND (Ht_mode(Run)=2) THEN
10815       Esum=(Edat(5,Ne)-Edat(5,P2n))+Esum
10820       Tsum=(Ne-P2n)*Ts+Tsum
10825       Tsum2=((Ne-P2n)*Ts)↑2+Tsum2
10830       Etsum=(Edat(5,Ne)-Edat(5,P2n))*(Ne-P2n)*Ts+Etsum
10835     END IF
10840 !        STORE IG EMF DATA
10845     Edat(1,Ne)=Vreading(1)
10850 !        PRINT TEMP. READINGS AND CONTROL OUTPUT ON CRT
10855     IF Screen_prnt THEN
10860       PRINT
10865       PRINT "  LOOP 1 : IG (microV) :";1.E+6*Edat(1,Ne);"   CONTROL OUTPUT :";Cdat
a(1,2,1)
10870       PRINT "  LOOP 2 : OG TEMP.(C) :";Edat(2,Ne);"   CONTROL OUTPUT :";Cdata(2,2,
1)
10875       PRINT "  LOOP 3 : TAHP TEMP.(C) :";Edat(3,Ne);"   CONTROL OUTPUT :";Cdata(3,
2,1)
10880       PRINT "  LOOP 4 : BAHP TEMP.(C) :";Edat(4,Ne);"   CONTROL OUTPUT :";Cdata(4,
2,1)
10885       PRINT "  LOOP 5 : MHP TEMP. (C) :";Edat(5,Ne);"   CONTROL OUTPUT :";Cdata(5,
2,1)
10890     END IF
10895     IF (Ne MOD (INT(300/Ts)))=0 OR (Ne=1) THEN   ! PRINT OUT TEMPS.
10900             ! EVERY 300 SECONDS
10905       T1=(Ts*Ne)/60.
10910 Fmt1:  IMAGE DDDDD.DD,5X,DDDD.DDD,5X,DDDD.DDD,5X,DDDD.DDD,5X,DDDD.DDD,5X,MDDDD.D
10915       OUTPUT 701 USING Fmt1;T1,Edat(5,Ne),Edat(2,Ne),Edat(3,Ne),Edat(4,Ne),Edat(1,
Ne)*1.E+6
10920     END IF
10925 !    UPDATE THE EXECUTION QUEUE (UNTIL THE 'FINISH' DATA SEQUENCE STARTS)
10930     IF Igflag=0 THEN
10935 !           MAIN HEATER CURRENT SCAN  (PHASE 1 AND PHASE 2)
10940       IF (Ne MOD Ntp)=0 THEN CALL Pack_queue(Queue(*),Nq,Qseq4(*),Nqs4)
10945 !          tc MEASUREMENTS & ISO.BLOCK TEMP. READING  (PHASE 2)
10950       IF Td0>0 AND (Ne MOD Ntm)=0 THEN CALL Pack_queue(Queue(*),Nq,Qseq1(*),Nqs1)
10955 !           DVM ZERO MEASUREMENTS   (PHASE 1)
10960       IF (Ne MOD Ntz)=0 AND Td0=0 THEN CALL Pack_queue(Queue(*),Nq,Qseq5(*),Nqs5)
10965 !          ISOBLOCK TEMP. MEASUREMENT  (PHASE 1)
10970       IF ((Ne MOD Ntr)=0) AND (Td0=0) THEN CALL Pack_queue(Queue(*),Nq,Qseq2(*),Nq
s2)
10975     END IF
10980     OFF TIMEOUT
10985     SUBEXIT
10990 Nothing:   !
10995     BEEP 83*23,.1
11000     RETURN
11005 Adj_error:   !
11010     IF ERRL(10915) THEN
11015       PRINTER IS 1
11020       PRINT "  PRINTER MALFUNCTION  ";ERRM$
11025       BEEP 83*30,.1
11030       GOTO 10920
11035     END IF
11040     PRINT "   ADJUST SUBROUTINE ERROR :"
11045     PRINT ERRM$
11050     CALL Err_record(ERRM$)
11055     BEEP 83*30,.2
11060  SUBEND
11065 ! ////////////////////////////////////////////////////////////////////////////
11070  SUB Pid(M,Ein,Sout)
11075 !   THIS SUB IS A DIGITAL CONTROLLER. GIVEN AN INPUT SIGNAL (Ein) IT WILL
11080 !   COMPUTE AN OUTPUT RESPONSE (Sout) FOR A GIVEN SET OF PID SETTINGS.
11085 !        Cset(i,j) : CONTAINS CONTROLLER SETPOINT,GAIN,Ti,Td,Ti BELL WIDTH
11090 !                    K BELL WIDTH, AND K REDUCTION FACTORS (INDEXED BY j).
```

```
11095 !                    THE INDEX i REFERS TO DIFFERENT CONTROL LOOPS.
11100 !        Cdata(i,j,k) : CONTAINS HISTORY OF CONTROL INPUT (j=1) AND
11105 !                       OUTPUT (j=2) OVER THREE INTERVALS (k=1 TO 3).
11110 !                       i HAS THE SAME MEANING AS IN Cset(*).
11115     SYSTEM PRIORITY 0
11120     COM /Adjustlocal/ Summ(*)
11125     COM /Ctr1/ Cdata(*),Cset(*),Cntrl_vlim(*),Loop_label$(*),Cstr$(*),Mhvmax
11130     COM /Sdisp/ Screen_prnt
11135     COM /Fl/ Flag$
11140     COM /Mc1/ Ts,Ne,Edat(*),Pdat(*),Nr,Rtdat(*),Nf,Fedat(*),Ftdat(*),Kdat(*),Tme(*)
,Nrmax,Nfmax,Tlim
11145     ON ERROR GOTO Pid_error
11150     FOR I=3 TO 2 STEP -1
11155        Cdata(M,2,I)=Cdata(M,2,I-1)
11160        Cdata(M,1,I)=Cdata(M,1,I-1)
11165     NEXT I
11170 !
11175     Cdata(M,1,1)=Cset(M,1)-Ein    !   MOST RECENT VALUE OF ERROR SIGNAL
11180 !
11185 !      ******    MODIFIED PID CONTROL SCHEME    ******
11190 !   GAIN BELL(1) IS THE WIDE BELL.  BELL(2) IS THE NARROW (NOISE) BELL.
11195 !   1 DEGREE = 0.6 mV (IG) = 0.375 OHM (MH, OG, T/BAH) = APPROX. 0.4 OHM
11200     Sp1=10          !  FOR RTD's PROP. GAIN BELL(1) IS 25K =10 OHMS WIDE.
11205     IF M=1 THEN Sp1=.006    !  FOR IG GAIN BELL(1) IS 10K = 6  mV  WIDE.
11210     Gain=Cset(M,2)
11215     Ti=Cset(M,3)
11220     Td=Cset(M,4)
11225     Tibw=Cset(M,5)          !  INTEGRATOR BELL(1) WIDTH
11230     Kbw=Cset(M,6)           !  GAIN BELL(2) WIDTH.
11235     Krf0=Cset(M,7)          !  INITIAL GAIN REDUCTION FACTOR
11240     Tdbw=.04                !  DERIVATIVE BELL WIDTH
11245     IF M=1 THEN Tdbw=6.E-5  !  Tdbw IS 0.1 DEG FOR ALL PLATES
11250 !
11255 !   ADJUSTMENTS OF CONTROL PARAMETERS FOR SETPOINTS AND GAIN.
11260     A1=0                    !  SLOPE OF  Ti DEPENDENCE ON T(R)
11265     A2=0                    !  SLOPE OF KRF DEPENDENCE ON T(R)
11270     Delr1=Cset(M,1)-100     ! SIMPLIFIED REF. PT. (LIMIT RUNS TO 273 K)
11275     IF M=1 THEN Delr1=Cset(5,1)-100
11280     Delr2=Cset(5,1)-Cset(4,1)/2-Cset(3,1)/2
11285     Delr0=7.5
11290     SELECT M
11295     CASE 1                  !  IG
11300        A1=.35
11305     CASE 2                  !  OG
11310        Delr2=Delr1
11315        A2=.00067
11320        Delr0=16
11325     CASE 3 TO 4             !  T/BAH
11330        A1=2.5
11335        Delr2=Delr1
11340        Delr0=12.3
11345     CASE 5                  !  MH
11350        A2=0
11355     END SELECT
11360     Ti=Ti+A1*Delr1
11365     Krf=Krf0+A2*Delr1
11370     IF Krf<1.E-10 THEN Krf=1.E-10
11375     Tibw=Tibw*Delr2*Krf0/(Krf*Delr0)
11380     Krf2=Krf
11385     IF M=5 THEN Krf2=Krf0-A2*Delr1+.02
11390     IF Krf2<0 THEN Krf2=0
11395 !
11400 !         PROPORTIONAL GAIN TERM
11405 !   Vpg = VARIABLE PROPORTIONAL GAIN FACTOR; COMMON TO ALL GAIN TERMS
11410     IF Kbw<1.E-10 THEN Kbw=1.E-10
11415     Arg1=(Cdata(M,1,1)/Sp1)↑2
11420     Arg2=(Cdata(M,1,1)/Kbw)↑2
11425     IF Arg1<20 THEN
11430        Vpg=1-(1-Krf)*EXP(-Arg1)
11435     ELSE
11440        Vpg=1          !     CORRECT TO WITHIN 2 ppb
11445     END IF
```

```
11450      Vpg1=Vpg*Gain      !     GAIN WITH BELL, WITHOUT FINGER; USED BELOW
11455      IF Arg2<20 THEN Vpg=Vpg*(1-(1-Krf2)*EXP(-Arg2))
11460   !              OTHERWISE Vpg APPROACHES ZERO
11465      Vpg=Vpg*Gain
11470      Propv=Vpg*Cdata(M,1,1)
11475 !
11480 !          INTEGRAL TERM
11485 !    Vig = VARIABLE INTEGRATOR FACTOR
11490      IF Tibw<1.E-10 THEN Tibw=1.E-10
11495      Arg3=(Cdata(M,1,1)/Tibw)↑2
11500      IF Arg3<20 THEN
11505         Vig=EXP(-Arg3)
11510      ELSE
11515         Vig=0
11520      END IF
11525      Vig=Vig*Vpg1          !   PROPORTIONAL GAIN TERM (WIDE BELL)
11530      Tpidsum=Summ(M)+Vig*Ts*Cdata(M,1,1)/(Ti+1.0E-10)
11535      IF Tpidsum<(Cntrl_v!im(M))↑2 THEN Summ(M)=Tpidsum
11540      IF Summ(M)<0 THEN Summ(M)=0
11545      Intv=Summ(M)
11550 !
11555 !          DERIVATIVE TERM
11560 !    Vdg = VARIABLE DERIVATIVE GAIN FACTOR
11565 !    CHOOSE SMALLER DIFFERENCE TO AVOID DISTURBANCE BY CURRENT ADJ.
11570      Cd12=Cdata(M,1,1)-Cdata(M,1,2)
11575      Cd23=Cdata(M,1,2)-Cdata(M,1,3)
11580      Cd13=Cdata(M,1,1)-Cdata(M,1,3)
11585      Diff0=.008
11590      Cdiff=Cd13/2
11595      IF (ABS(Cd12-Cd23)>Diff0) AND (ABS(Cd12)>ABS(Cd23)) THEN
11600         Cdiff=Cd23
11605      ELSE
11610         Cdiff=Cd12
11615      END IF
11620      Arg4=(Cdata(M,1,1)/Tdbw)↑2
11625      IF Arg4<20 THEN
11630         Vdg=1-EXP(-Arg4)
11635      ELSE
11640         Vdg=1
11645      END IF
11650      Vdg=Vdg*Td*Cdiff/Ts
11655      Deriv=Vdg*Vpg1                !   PROPORTIONAL GAIN TERM (WIDE BELL)
11660      IF Ne<3 THEN Deriv=-Propv   !   AVOID SINGULARITIES ON START-UP
11665 !          TOTAL PID SIGNAL
11670      IF Screen_prnt THEN PRINT "LOOP=";M;"   PROP=";Propv;"   INT=";Intv;"   DERIV="
;Deriv
11675      Cs=Propv+Intv+Deriv
11680 !
11685 !        ******      END OF CONTROL SCHEME      ******
11690      IF Cs<0. THEN Cs=0.
11695 !       SQUARE ROOT TAKEN TO LINEARIZE OUTPUT TO POWER (IE. V↑2)
11700      Cs=SQR(Cs)
11705      IF Cs>Cntrl_vlim(M) THEN Cs=Cntrl_vlim(M)    !  LIMIT VOLTAGE OUTPUT
11710      Cdata(M,2,1)=Cs
11715      Sout=Cs
11720      SUBEXIT
11725 Pid_error:   !
11730      Cdata(M,2,1)=0
11735      PRINT "   PID SUBROUTINE ERROR : "&ERRM$
11740      BEEP 83*32,.2
11745      CALL Err_record(ERRM$)
11750  SUBEND
11755 ! ////////////////////////////////////////////////////////////////////////
11760  SUB Plot_switch
11765   !       THIS SUB ALLOWS THE USER TO PERFORM A RUNTIME 'SWITCH' OF THE
11770   !       DISPLAYED PLOT AND ITS X-AXIS RANGE.
11775      COM /Mc1/ Ts,Ne,Edat(*),Pdat(*),Nr,Rtdat(*),Nf,Fedat(*),Ftdat(*),Kdat(*),Tme(*)
,Nrmax,Nfmax,Tlim
11780      COM /Flgs/ Igflag,Prev_ne
11785      COM /Gr1/ Plot_view,Plot_type,Pindex
11790      COM /Ioscan/ Queue(*),Nq,Qseq1(*),Nqs1,Qseq2(*),Nqs2,Qseq3(*),Nqs3,Qseq4(*),Nqs
4,Qseq5(*),Nqs5,Qseq6(*),Nqs6
```

```
11795     COM /Gr2/ X1,X2,Xinc,Y1,Y2,Yinc,Xtit$,Ytit$
11800     COM /Sb1/ T0,Td0
11805     SYSTEM PRIORITY 0
11810     ON ERROR GOTO Plot_sw_error
11815     IF Queue(1)>-1 THEN
11820        BEEP 83*23,.1
11825        SUBEXIT
11830     END IF
11835     DISP ""
11840     Beat_it=0
11845     Tpindex=Pindex
11850     Tplot_type=Plot_type
11855     Tplot_view=Plot_view
11860     FOR I=0 TO 9
11865        ON KEY I LABEL "" GOTO 12135
11870     NEXT I
11875     ON KEY 0 LABEL "SELECT DATA" GOTO 11945
11880     ON KEY 2 LABEL "SELECT VIEW" GOTO 12050
11885     ON KEY 8 LABEL "  PLOT IT  " GOTO 11910
11890     ON KEY 9 LABEL " MAIN MENU " GOTO 11900
11895     GOTO 12035
11900     Beat_it=1
11905     GOTO 11925
11910     Plot_type=Tplot_type
11915     Pindex=Tpindex
11920     Plot_view=Tplot_view
11925     CALL Plot_prep(X1,X2,Xinc,Y1,Y2,Yinc,Xtit$,Ytit$)
11930     CALL Rescale_plot(X1+(X2-X1)/2,Y1+(Y2-Y1)/2)
11935     IF Beat_it THEN SUBEXIT
11940     GOTO 12035
11945     Id=Id+1
11950     IF Id>12 THEN Id=1
11955     IF Tme(1)=0 AND ((Id>5 AND Id<10) OR Id=12) THEN GOTO 11945
11960     Tpindex=1
11965     SELECT Id
11970     CASE 1 TO 5
11975        Tplot_type=1
11980        Tpindex=Id
11985     CASE 6 TO 9
11990        Tplot_type=2
11995        Tpindex=Id-5
12000     CASE 10
12005        Tplot_type=3
12010     CASE 11
12015        Tplot_type=4
12020     CASE 12
12025        Tplot_type=5
12030     END SELECT
12035     CALL G_label(Glab$,Tplot_type,Tpindex)
12040     ON KEY 5 LABEL Glab$ GOTO 12135
12045     GOTO 12060
12050     Tplot_view=Tplot_view+1
12055     IF Tplot_view>5 THEN Tplot_view=0
12060     SELECT Tplot_view
12065     CASE 1
12070        Pv$="LAST 30 MIN."
12075     CASE 0
12080        Pv$=" FULL  VIEW"
12085     CASE 2
12090        Pv$="LAST 60 MIN."
12095     CASE 3
12100        Pv$="LAST 120 MIN."
12105     CASE 4
12110        Pv$="LAST 240 MIN."
12115     CASE 5
12120        Pv$="LAST 480 MIN."
12125     END SELECT
12130     ON KEY 7 LABEL Pv$ GOTO 12135
12135     Tp=TIMEDATE
12140     LOOP
12145        IF Ne<>Prev_ne THEN CALL Update_plot("N")
12150     EXIT IF TIMEDATE-Tp>120
```

```
12155        IF Queue(1)>-1 THEN
12160           BEEP 83*23,.1
12165           SUBEXIT
12170        END IF
12175     END LOOP
12180     IF Beat_it=0 THEN GOTO 11900
12185     SUBEXIT
12190 Plot_sw_error:   !
12195     PRINT "PLOT SWITCH ERROR  ";ERRM$
12200     CALL Err_record(ERRM$)
12205     OFF ERROR
12210  SUBEND
12215 ! ////////////////////////////////////////////////////////////////////////
12220  SUB Pblank(X1,X2,Xinc,Y1,Y2,Yinc,Xtit$,Ytit$)
12225   !   THIS SUB DRAWS A BLANK PLOT, WITH LABELED AXES AND A GRID
12230     COM /Dt1/ File_specs$(*),Mode$(*),Gas$(*)
12235     COM /Gr1/ Plot_view,Plot_type,Pindex
12240     COM /Mc5/ Op_mode(*),Htemp(*),Ltemp(*),Ht_mode(*),File_num(*),Set_temp(*)
12245     COM /Rtd_corr/ Tcorr_rtd,Rtd_adj_flag,Sp_corr(*)
12250     ON ERROR GOTO Pblank_error
12255     GINIT
12260     GRAPHICS ON
12265     VIEWPORT 20,127,24,92
12270     WINDOW X1,X2,Y1,Y2
12275     FRAME
12280     LINE TYPE 4,5
12285     GRID Xinc,Yinc,X1,Y1
12290     LINE TYPE 1
12295     VIEWPORT 0,130,0,100
12300     LORG 5
12305     CSIZE 2.5
12310     Xl=X1-.06*(X2-X1)
12315     Yl=Y1-.08*(Y2-Y1)
12320     FOR X=X1 TO X2+Xinc/10. STEP Xinc
12325        MOVE X,Yl
12330        LABEL VAL$((INT(X/6))/10)
12335     NEXT X
12340     FOR Y=Y1 TO Y2+Yinc/10. STEP Yinc
12345        MOVE Xl,Y
12350        Yp=Y
12355        IF ABS(Yp)<1.E-10 THEN Yp=0
12360        LABEL VAL$(Yp)
12365     NEXT Y
12370     MOVE X1+(X2-X1)/2,Y1-.14*(Y2-Y1)
12375     CSIZE 3.2
12380     LABEL Xtit$
12385     DEG
12390     LDIR 90
12395     MOVE X1-.15*(X2-X1),Y1+(Y2-Y1)/2
12400     LABEL Ytit$
12405     LDIR 0
12410     LORG 2
12415     MOVE X1-.15*(X2-X1),Y1-.16*(Y2-Y1)
12420     CSIZE 2.2
12425     LABEL "FILE NAME : "&File_specs$(1)[1,10]
12430     MOVE X1-.15*(X2-X1),Y1-.19*(Y2-Y1)
12435     LABEL "RUN DATE : "&File_specs$(1)[11,31]
12440     VIEWPORT 20,127,24,92
12445     IF Plot_type=1 AND Pindex>2 THEN
12450        MOVE X1+.02*(X2-X1),Y1+.02*(Y2-Y1)
12455        LORG 1
12460        CSIZE 2.4
12465        Correc=-INT(Sp_corr(Pindex)*10000.)/10000.
12470        Spoff=INT((Set_temp(Pindex)-Sp_corr(Pindex))*10000.)/10000.
12475        Spunc=INT(Set_temp(Pindex)*10000.)/10000.
12480        LABEL "S.P. (degC) = "&VAL$(Spunc)&" ,  S.P. (W/OFFSET) = "&VAL$(Spoff)&" ,
  OFFSET = "&VAL$(Correc)
12485     END IF
12490     MOVE X1-.18*(X2-X1),Y1+.5*(Y2-Y1)
12495     SUBEXIT
12500 Pblank_error:   !
12505     PRINT "PBLANK ERROR  ";ERRM$
```

```
12510      CALL Err_record(ERRM$)
12515  SUBEND
12520 !////////////////////////////////////////////////////////////////////////
12525  DEF FNOhms_rtd(Rtd_num,T)
12530 !  THIS ROUTINE CONVERTS TEMPERATURE (C)  TO OHMS
12535 !  FOR THE ROSEMOUNT PLATINUM RTD'S USING THE ITPS-68 FORM OF THE
12540 !  CALLENDAR-VAN DUSEN EQUATION.
12545      COM /Canv_dat/ R0(*),Alpha(*),Delta(*)
12550      ON ERROR GOTO Ohms_rtd_errar
12555      R=R0(Rtd_num)
12560      A=Alpha(Rtd_num)
12565      D=Delta(Rtd_num)
12570 !    CONVERT TEMP. (C) TO RESISTANCE (OHMS)
12575      T2=T-.045*(T/100)*(T/100-1)*(T/419.58-1)*(T/630.74-1)
12580      Resistance=R*(1+A*(T2-D*(T2/100)*(T2/100-1)))
12585      OFF ERROR
12590      RETURN Resistance
12595 Ohms_rtd_errar:    !
12600      PRINT "RTD CONVERSION ERROR : ";ERRM$
12605      CALL Err_record(ERRM$)
12610  FNEND
12615 !////////////////////////////////////////////////////////////////////////
12620  DEF FNChan_sig(Ch)
12625 !      THIS SUB CALCULATES AN INTEGER COMMAND FOR THE DIGITAL
12630 !      OUTPUT CARD IN THE H.P. MULTIPROGRAMMER THAT WILL SET
12635 !      THE JULIE RELAY TO THE CHANNEL 'Ch'.
12640      ON ERROR GOTO Chan_sig_err
12645      IF Ch<10 THEN
12650         N=240+Ch
12655      ELSE
12660         N=15+16*(Ch-10)
12665      END IF
12670      IF N<0 THEN
12675         N=0
12680         PRINT " REQUESTED JULIE RELAY CHANNEL IS OUT OF RANGE!!"
12685      END IF
12690      IF N>255 THEN
12695         N=255
12700         PRINT " REQUESTED JULIE RELAY CHANNEL IS OUT OF RANGE!!"
12705      END IF
12710      RETURN N
12715 Chan_sig_err:     !
12720      PRINT "  ERROR IN SUB 'CHAN_SIG'"
12725      CALL Err_record(ERRM$)
12730  FNEND
12735 !////////////////////////////////////////////////////////////////////////
12740  SUB Rescale_plot(X,Y)
12745 !      THIS SUB RESCALES PLOTS THAT HAVE GONE OFFSCALE
12750      COM /Gr1/ Plot_view,Plot_type,Pindex
12755      COM /Gr2/ X1,X2,Xinc,Y1,Y2,Yinc,Xtit$,Ytit$
12760      COM /Mc1/ Ts,Ne,Edat(*),Pdat(*),Nr,Rtdat(*),Nf,Fedat(*),Ftdat(*),Kdat(*),Tme(*)
,Nrmax,Nfmax,Tlim
12765      ON ERROR GOTO Rescale_errar
12770      IF Y<Y1 THEN Y1=Y1-Yinc
12775      IF Y>Y2 THEN Y2=Y2+Yinc
12780      IF (Y2-Y1)/Yinc>10 THEN CALL Plat_prep(X1,X2,Xinc,Y1,Y2,Yinc,Xtit$,Ytit$)
12785      IF X>(X1+.9*(X2-X1)) THEN
12790         Xrange=X2-X1
12795         IF Plat_view>=1 THEN
12800            X2=X2+Xinc
12805            X1=X2-Xrange
12810         ELSE
12815            X2=X2+Xrange
12820            Xinc=Xinc*2
12825         END IF
12830      END IF
12835      CALL Pblank(X1,X2,Xinc,Y1,Y2,Yinc,Xtit$,Ytit$)
12840      SELECT Plat_type
12845      CASE 1,3
12850         IF Plot_view>=1 THEN Nstart=INT(X1/Ts)
12855         IF Nstart=0 THEN Nstart=1
12860         Nend=Ne
```

```
12865      CASE 2
12870         Nstart=1
12875         Nend=Nf
12880         IF Tme(Nf)=0 THEN Nend=Nf-1
12885         IF Nend<1 THEN Nend=1
12890      CASE 4
12895         Nstart=1
12900         Nend=Nr
12905      CASE 5
12910         Nstart=1
12915         Nend=Nf
12920         IF Kdat(Nf)=0 THEN Nend=Nf-1
12925         IF Nend<1 THEN Nend=1
12930      CASE ELSE
12935      END SELECT
12940      FOR I=Nstart TO Nend
12945         SELECT Plot_type
12950         CASE 1
12955            Xt=I*Ts
12960            Yt=Edat(Pindex,I)
12965            IF Pindex=1 THEN Yt=Yt*1.E+6
12970         CASE 2
12975            Xt=Tme(I)
12980            Yt=Ftdat(1+(Pindex-1)*2,I)+Ftdat(2+(Pindex-1)*2,I)
12985         CASE 3
12990            Xt=I*Ts
12995            Yt=Pdat(1,I)*Pdat(2,I)
13000         CASE 4
13005            Xt=Rtdat(2,I)
13010            Yt=Rtdat(1,I)
13015         CASE 5
13020            Xt=Tme(I)
13025            Yt=Kdat(I)
13030         CASE ELSE
13035         END SELECT
13040         IF (Plot_type=2 OR Plot_type=5) THEN
13045            IF I=1 THEN MOVE Xt,Yt
13050            IF I>1 THEN
13055               IF Tme(Nf)<>0 THEN PLOT Xt,Yt
13060            END IF
13065         ELSE
13070            PLOT Xt,Yt
13075         END IF
13080      NEXT I
13085      SUBEXIT
13090 Rescale_error:   !
13095      PRINT " RESCALE PLOTTING ERROR  ";ERRM$
13100      CALL Err_record(ERRM$)
13105   SUBEND
13110 !///////////////////////////////////////////////////////////////////////
13115   SUB G_label(G$,Plot_type,P)
13120 ! THIS SUB RETURNS A LABEL FOR THE CRT DATA PLOTS
13125      ON ERROR GOTO G_label_err
13130      SELECT Plot_type
13135      CASE 1
13140         IF P=1 THEN G$="    IG   tc"
13145         IF P=2 THEN G$="    OG   RTD"
13150         IF P=3 THEN G$="    TAP  RTD"
13155         IF P=4 THEN G$="    BAP  RTD"
13160         IF P=5 THEN G$="    MHP  RTD"
13165      CASE 2
13170         IF P=1 THEN G$="   TOP   tc"
13175         IF P=2 THEN G$=" U.MAIN  tc"
13180         IF P=3 THEN G$=" L.MAIN  tc"
13185         IF P=4 THEN G$=" BOTTOM  tc"
13190      CASE 3
13195         G$="  MAIN POWER"
13200      CASE 4
13205         G$="  REF. BLOCK"
13210      CASE 5
13215         G$=" THERM. COND."
13220      END SELECT
```

```
13225     SUBEXIT
13230 G_label_err:    !
13235     PRINT "   ERROR IN SUB 'G_LABEL'"
13240     CALL Err_record(ERRM$)
13245  SUBEND
13250 !//////////////////////////////////////////////////////////////////////////
13255  SUB Read_io(V,Tread,Instrument,Dvm_cmd$,Oldvolt)
13260   !    THIS SUB TAKES AN INSTRUMENT READING & TRAPS INSTRUMENT READ
13265   !    ERRORS.
13270     COM /Instr/ Mh181,Jrl181,Mh195,Tap195,Bap195,Ig195
13275     COM /Read1/ Io_error,Bad_instr(*),Bad_read_time$(*)
13280     COM /Sb1/ T0,Td0
13285     COM /Fl/ Flag$
13290     COM /Zeros/ Zjrl181_200,Zjrl181_20,Zgap195,Zhistory(*)
13295     Error=0
13300     ON TIMEOUT 7,1. GOTO Read_it_error
13305     ON ERROR GOTO Read_io_err
13310     Zero=0
13315     IF Instrument=Ig195 THEN Zero=Zgap195
13320     IF Instrument=Jrl181 THEN
13325        Zero=Zjrl181_200
13330        IF POS(Dvm_cmd$,"R2")<>0 THEN Zero=Zjrl181_20
13335     END IF
13340     Tt=TIMEDATE
13345     ENTER Instrument;V
13350     V=V-Zero
13355     Tread=TIMEDATE-T0
13360     OFF TIMEOUT
13365     SUBEXIT
13370 Read_it_error:   !
13375     ON TIMEOUT 7,1. GOTO Read_it_error
13380     Error=Error+1
13385     Io_error=Io_error+1
13390     Bad_instr(Io_error)=Instrument
13395     Bad_read_time$(Io_error)=DATE$(TIMEDATE)&" "&TIME$(TIMEDATE)
13400     PRINT "   IO ERROR IN SUB 'READ_IO'"
13405     BEEP 83*35,.1
13410     IF Io_error=100 THEN
13415        Flag$="MAXIMUM NUMBER OF IO ERRORS EXCEEDED !!"
13420        SUBEXIT
13425     END IF
13430     IF Error=1 THEN
13435        CLEAR Instrument
13440        WAIT .2
13445        BEEP 83,.1
13450        OUTPUT Instrument;Dvm_cmd$
13455        WAIT .2
13460        GOTO 13300
13465     ELSE
13470        V=Oldvolt
13475        Tread=TIMEDATE-T0
13480     END IF
13485     SUBEXIT
13490 Read_io_err:    !
13495     PRINT "   ERROR IN SUB 'READ_IO'"&ERRM$
13500     CALL Err_record(ERRM$)
13505  SUBEND
13510 !//////////////////////////////////////////////////////////////////////////
13515  SUB Tc_store(Chan0,V,Tm)
13520   !      THIS SUB STORES T.C. VOLTAGE DATA (PHASE 2)
13525     COM /Mc1/ Ts,Ne,Edat(*),Pdat(*),Nr,Rtdat(*),Nf,Fedat(*),Ftdat(*),Kdat(*),Tme(*)
,Nrmax,Nfmax,Tlim
13530     COM /Sb2/ I_rtd,Tref,Emf_ref
13535     COM /Sb3/ Fd(*),Tavg_interval
13540     COM /Tcst1/ Store_flag
13545     ON ERROR GOTO Tc_store_error
13550     IF Store_flag=0 THEN
13555        Nf=Nf+1
13560        IF Nf>Nfmax THEN
13565           Nf=Nfmax
13570           PRINT "   DATA STORAGE ARRAYS ARE FULL!"
13575           BEEP 83*5,.1
```

```
13580           Flag$="   OVERFLOW ON FINAL DATA STORAGE ARRAYS!"
13585           SUBEXIT
13590        END IF
13595        Store_flag=1
13600     END IF
13605     IF Chan0<10 THEN
13610 !            STORE THE EMF AND TEMP. DATA
13615        Indx=2*Chan0-5
13620        IF Chan0=3 AND Fedat(1,Nf)=0 THEN Store_flag=Tm
13625        IF Chan0=3 AND Fedat(1,Nf)<>0 THEN
13630           Tme(Nf)=(Store_flag+Tm)/2
13635           Store_flag=0
13640        END IF
13645        IF Fedat(Indx,Nf)=0 THEN
13650           Fedat(Indx,Nf)=V
13655        ELSE
13660           Fedat(Indx,Nf)=(Fedat(Indx,Nf)+V)/2
13665        END IF
13670     ELSE
13675 !                STORE DELTA-EMF DATA
13680        Indx=(Chan0-12)*2
13685        IF Fedat(Indx,Nf)=0 THEN
13690           Fedat(Indx,Nf)=V
13695        ELSE
13700           Fedat(Indx,Nf)=(Fedat(Indx,Nf)+V)/2
13705        END IF
13710     END IF
13715     SUBEXIT
13720 Tc_store_error:   !
13725     PRINT " TC_STORE ERROR  ";ERRM$
13730     CALL Err_record(ERRM$)
13735  SUBEND
13740 !///////////////////////////////////////////////////////////////////////////
13745  SUB K_ghp(K,Tlo,Thi,Dxc,Acor,T1,T2,T3,T4,Q,Dia,Dx,Dr,Sc,Rc,Ac)
13750 !    THIS SUB COMPUTES THERMAL CONDUCTIVITY FOR A GUARDED HOT PLATE
13755 !    SPECIMEN. VARIABLES ARE DEFINED AS FOLLOWS :
13760 !        INPUT :
13765 !           T1 = AVG. LOWER AUX. PLATE TEMP.    (C)
13770 !           T2 = AVG. LOWER MAIN PLATE TEMP.    (C)
13775 !           T3 = AVG. UPPER MAIN PLATE TEMP.    (C)
13780 !           T4 = AVG. UPPER AUX. PLATE TEMP.    (C)
13785 !            Q = AVG. POWER INPUT TO METERED AREA (mW)
13790 !          Dia = MAIN PLATE DIAMETER (cm)
13795 !           Dx = SAMPLE THICKNESS (cm)
13800 !           Dr = MAIN-INNER GUARD GAP WIDTH (cm)
13805 !           Sc = PLATE SPACER CODE (1=QUARTZ, 2=St.St.)
13810 !           Rc = RUN MODE CODE (1=DBLE SIDED, 2=TOP, 3=BOT.)
13815 !           Ac = APPARATUS CODE (0=HIGH TEMP., 1=LOW TEMP.)
13820 !       OUTPUT :
13825 !         Acor = THERMAL EXPANSION AND GAP CORRECTED PLATE AREA (cm↑2)
13830 !          Dxc = CORRECTED SAMPLE THICKNESS (cm)
13835 !            K = THERMAL CONDUCTIVITY (mW/(m*K))
13840 !          Tlo = AVG. LOW TEMP. OF THE PLATES    (C)
13845 !          Thi = AVG. HIGH TEMP. OF THE PLATES   (C)
13850 !
13855 !       OTHER VARIABLES :
13860 !         Diac = THERMAL EXPANSION CORRECTED PLATE DIAMETER (cm)
13865 !         Diac_wg = T. EXPANSION AND GAP CORRECTED PLATE DIAMETER (cm)
13870     ON ERROR GOTO K_ghp_error
13875  !    CALCULATE TEMP. DIFF. ACROSS SPECIMEN
13880     SELECT Rc
13885     CASE 1
13890        Tlo=(T1+T4)/2
13895        Thi=(T2+T3)/2
13900     CASE 2
13905        Tlo=T4
13910        Thi=T3+(T2-T1)
13915     CASE 3
13920        Tlo=T1
13925        Thi=T2+(T3-T4)
13930     END SELECT
13935     Tbar=(Tlo+Thi)/2+273.15
```

```
13940     Delt=T3-T4+T2-T1
13945     Tp=(T2+T3)/2
13950  !    ALUMINA THERMAL EXPANSION CORRECTION ON DIAMETER OF THE MAIN PLATE
13955     Diac=Dia*(1+7.5E-6*(Tp-20))
13960  !    AREA CORRECTION FOR GAP
13965     Diac_wg=2*Dr*(1+7.5E-6*(Tp-20))+Diac     !  DIA. OF THE PLATE + GAP
13970     Ap=PI*(Diac↑2/4)                          !  AREA OF THE PLATE
13975     Agap=PI*(Diac_wg↑2/4)-Ap                  !  AREA OF THE GAP
13980     Acor=Ap+Agap/2                            !  CORRECTED AREA
13985  !     THERMAL EXPANSION CORRECTION FOR THICKNESS SPACERS
13990     SELECT Sc
13995     CASE 1                                    !  QUARTZ
14000        Dxc=Dx*(1+1.7E-5*(Tbar-20))
14005     CASE 2                                    !  STAINLESS STEEL
14010        Dxc=Dx+(Dx/100.)*(-.358+9.472E-4*Tbar+1.031E-6*Tbar↑2-2.978E-10*Tbar↑3)
14015     CASE ELSE
14020        Dxc=Dx
14025     END SELECT
14030  !       IT MAY BE NECESSARY TO CORRECT FOR PLATE SAG IN THE SINGLE SIDE
14035  !       RUNS. Dxc IS LARGER FOR HEAT FLOW UP AND SMALLER FOR HEAT FLOW
14040  !       DOWN. NOT KNOWN AT THIS TIME - NOT NECESSARY FOR RIGID SPECIMENS.
14045  !               CALCULATE THERMAL CONDUCTIVITY
14050     K=100.*Q*Dxc/(Acor*Delt)
14055     SUBEXIT
14060 K_ghp_error:    !
14065     K=-1
14070     DISP "  CALCULATION ERROR   ";ERRM$
14075     CALL Err_record(ERRM$)
14080  SUBEND
14085 !///////////////////////////////////////////////////////////////////
14090  SUB Set_pnt_calc
14095 !       THIS SUB RECALCULATES SETPOINTS FOR THE CONTROL LOOPS. THE VALUES
14100 !       OF TEMP. IN THE 'Set_temp' ARRAY ARE USED AS TEMP. SETPOINTS.
14105     COM /Ctr1/ Cdata(*),Cset(*),Cntrl_vlim(*),Loop_label$(*),Cstr$(*),Mhvmax
14110     COM /Mc5/ Op_mode(*),Htemp(*),Ltemp(*),Ht_mode(*),File_num(*),Set_temp(*)
14115     COM /Rn/ Run
14120     COM /Rtd_corr/ Tcorr_rtd,Rtd_adj_flag,Sp_corr(*)
14125     ON ERROR GOTO Sp_err
14130    !    CALC SETPOINTS  (IN OHMS)
14135     FOR I=2 TO 5
14140        Cset(I,1)=FNOhms_rtd(I,(Set_temp(I)-Sp_corr(I)))
14145     NEXT I
14150    !    CALC IG SETPOINT (IN VOLTS)
14155     Cset(1,1)=Set_temp(1)/1.E+6
14160     IF Htemp(Run)<>Set_temp(5) THEN Htemp(Run)=Set_temp(5)
14165     Lowest=Set_temp(3)
14170     IF Set_temp(4)<Lowest THEN Lowest=Set_temp(4)
14175     IF Lowest<>Ltemp(Run) THEN Ltemp(Run)=Lowest
14180     SUBEXIT
14185 Sp_err:    !
14190     PRINT "  ERROR IN SUB 'SET_PNT_CALC' : "&ERRM$
14195     CALL Err_record(ERRM$)
14200  SUBEND
14205 !////////////////////////////////////////////////////////////////////////
14210  SUB Write_io(Instr,Cmd$)
14215  !       THIS SUB OUTPUTS COMMANDS TO INSTRUMENTS ON THE HPIB BUS
14220     COM /Fl/ Flag$
14225     COM /Instr/ Mh181,Jrl181,Mh195,Tap195,Bap195,Ig195
14230     COM /Jrlchan/ Chan,Tchan,Dvm_cmmd$,Default_chan
14235     COM /Read1/ Io_error,Bad_instr(*),Bad_read_time$(*)
14240     Error=0
14245     ON ERROR GOTO Write_err
14250     ON TIMEOUT 7,1. GOTO Wrt_time_err
14255     FOR Isis=1 TO 2
14260        OUTPUT Instr;Cmd$
14265        IF Instr<>723 THEN ENTER Instr;Dummy
14270     NEXT Isis
14275     IF Instr=Jrl181 THEN Dvm_cmmd$=Cmd$
14280     OFF TIMEOUT
14285     SUBEXIT
14290 Wrt_time_err:   !
14295     Error=Error+1
```

```
14300     Io_error=Io_error+1
14305     Bad_instr(Io_error)=Instr
14310     Bad_read_time$(Io_error)=DATE$(TIMEDATE)&" "&TIME$(TIMEDATE)
14315     BEEP 83*35,.1
14320     PRINT "  OUTPUT ERROR OCCURED ON HPIB ADDRESS ";Instr
14325     IF Instr=Jrl181 THEN PRINT "   JRL CHANNEL NUMBER ";Chan
14330     IF Instr=723 THEN Flag$="OUTPUT ERROR TO MULTIPROGRAMMER"
14335     IF Io_error=100 THEN Flag$="MAXIMUM NUMBER OF I.O. ERRORS EXCEEDED !!"
14340     IF Error=1 AND Instr<>723 THEN
14345        CLEAR Instr
14350        WAIT .2
14355        GOTO 14250
14360     END IF
14365     SUBEXIT
14370 Write_err:    !
14375     PRINT "  ERROR IN SUB 'WRITE_IO' : "&ERRM$
14380     CALL Err_record(ERRM$)
14385  SUBEND
14390 !//////////////////////////////////////////////////////////////////////
14395  SUB Outseven(Volts)
14400  !      THIS SUB SETS THE MAIN HEATER PLATE KEPCO POWER SUPPLY VOLTAGE
14405     ON ERROR GOTO Outseven_err
14410     Hvolts=INT(204.7*Volts)/204.7
14415     Lvolts=Volts-Hvolts
14420     Digit=INT(255-(255*Lvolts/.005))
14425     IF Digit<0 THEN Digit=0
14430     IF Digit>255 THEN Digit=255
14435     CALL Write_io(723,"OP,7,"&VAL$(Hvolts)&"T")
14440     WAIT .1
14445     CALL Write_io(723,"OP,5,"&VAL$(Digit)&"T")
14450     SUBEXIT
14455 Outseven_err:       !
14460     PRINT "  ERROR IN SUB 'OUTSEVEN' : "&ERRM$
14465     CALL Err_record(ERRM$)
14470  SUBEND
14475 !///////////////////////////////////////////////////////////////////////
14480  SUB Linear(X(*),Y(*),N,A,B,Mean,Sd)
14485 !       THIS SUB PERFORMS A LINEAR REGRESSION ON A PAIR OF DATA ARRAYS
14490 !       AND A MEAN AND STD.DEV. CALC. ON THE Y DATA ARRAY.
14495 !   X and Y  ARE THE ABSCISSA AND ORDINATE DATA ARRAYS
14500 !   N  IS THE NUMBER OF DATA POINTS
14505 !   Mean IS THE MEAN VALUE OF THE Y DATA ARRAY
14510 !   Var  IS THE VARIANCE OF THE Y DATA ARRAY
14515 !   Sd  IS THE STANDARD DEVIATION OF THE Y DATA ARRAY
14520 !   A and B  ARE THE INTERCEPT AND SLOPE OF THE LINEAR REGRESSION LINE.
14525 !             CURVE FIT MODEL :  Y = A + B*X
14530 !
14535     ON ERROR GOTO Lin_err
14540     ALLOCATE Tem(N)
14545     X1=0
14550     X2=0
14555     Y1=0
14560     Y2=0
14565     Z1=0
14570     A=0
14575     B=0
14580     X1=SUM(X)
14585     Y1=SUM(Y)
14590     MAT Tem= X*X
14595     X2=SUM(Tem)
14600     MAT Tem= Y*Y
14605     Y2=SUM(Tem)
14610     MAT Tem= X*Y
14615     Z1=SUM(Tem)
14620     Mean=Y1/N
14625     Var=(N*Y2-Y1↑2)/(N*(N-1))
14630     Sd=SQR(ABS(Var))
14635     Xb=X1/N
14640     Yb=Y1/N
14645     B=(Z1-N*Xb*Yb)/(X2-N*Xb*Xb)
14650     A=Yb-B*Xb
14655     SUBEXIT
```

```
14660 Lin_err:   !
14665     IF B=0 THEN B=1
14670     PRINT "  ERROR IN SUB 'LINEAR' : "&ERRM$
14675     CALL Err_record(ERRM$)
14680  SUBEND
14685 !//////////////////////////////////////////////////////////////////////
14690  SUB Manual
14695  !      THIS SUB ALLOWS THE USER TO PERFORM A 'RUN TIME' ADJUSTMENT
14700  !      OF THE PID CONTROLLER PARAMETERS.
14705     OPTION BASE 1
14710     COM /Ctr1/ Cdata(*),Cset(*),Cntrl_vlim(*),Loop_label$(*),Cstr$(*),Mhvmax
14715     COM /Flgs/ Igflag,Prev_ne
14720     COM /Ioscan/ Queue(*),Nq,Qseq1(*),Nqs1,Qseq2(*),Nqs2,Qseq3(*),Nqs3,Qseq4(*),Nqs
4,Qseq5(*),Nqs5,Qseq6(*),Nqs6
14725     COM /Manual/ Powerflag,Vreading(*)
14730     COM /Mc1/ Ts,Ne,Edat(*),Pdat(*),Nr,Rtdat(*),Nf,Fedat(*),Ftdat(*),Kdat(*),Tme(*)
,Nrmax,Nfmax,Tlim
14735     COM /Mc5/ Op_mode(*),Htemp(*),Ltemp(*),Ht_mode(*),File_num(*),Set_temp(*)
14740     SYSTEM PRIORITY 0
14745     ON ERROR GOTO Manual_err
14750     IF Queue(1)>-1 THEN
14755        BEEP 83*23,.1
14760        SUBEXIT
14765     END IF
14770     Ncpar=7      !     NUMBER OF CONTROLLER PARAMETERS
14775     Ncloops=5     !     NUMBER OF CONTROL LOOPS
14780     ALLOCATE Ctemp(1:Ncloops,1:Ncpar),Sptemp(1:Ncloops),C(1:Ncpar)
14785     MAT Ctemp= Cset
14790     FOR I=1 TO Ncloops
14795        Ctemp(I,1)=Set_temp(I)
14800     NEXT I
14805     Ctemp(1,5)=Cset(1,5)*1.E+6
14810     Ctemp(1,6)=Cset(1,6)*1.E+6
14815     Lindex=5
14820     Cindex=1
14825     Inc_pwr=0
14830     FOR I=0 TO 9
14835        ON KEY I LABEL "" GOTO 15190
14840     NEXT I
14845     ON KEY 2 LABEL "    STEP  UP   " GOTO 14950
14850     ON KEY 3 LABEL " POWER SUPPLY" GOTO 14875
14855     ON KEY 4 LABEL " MAKE CHANGES" GOTO 14970
14860     ON KEY 7 LABEL "    STEP DOWN " GOTO 14960
14865     ON KEY 9 LABEL "  MAIN MENU   " GOTO 15245
14870     GOTO 15020
14875     IF Powerflag=1 THEN
14880        CALL Poweroff
14885     ELSE
14890        CALL Poweron
14895     END IF
14900     GOTO 15020
14905     Cindex=Cindex+1
14910     IF Cindex>Ncpar THEN Cindex=1
14915     GOTO 15020
14920     Lindex=Lindex+1
14925     IF Lindex>Ncloops THEN Lindex=1
14930     GOTO 15020
14935     Inc_pwr=Inc_pwr+1
14940     IF Inc_pwr>10 THEN Inc_pwr=-10
14945     GOTO 15020
14950     Ctemp(Lindex,Cindex)=Ctemp(Lindex,Cindex)+10↑Inc_pwr
14955     GOTO 15020
14960     Ctemp(Lindex,Cindex)=Ctemp(Lindex,Cindex)-10↑Inc_pwr
14965     GOTO 15020
14970     FOR I=1 TO Ncloops
14975        FOR J=2 TO Ncpar
14980           Cset(I,J)=Ctemp(I,J)
14985        NEXT J
14990        Set_temp(I)=Ctemp(I,1)
14995     NEXT I
15000     Cset(1,5)=Ctemp(1,5)*1.E-6
15005     Cset(1,6)=Ctemp(1,6)*1.E-6
```

```
15010      CALL Set_pnt_calc       !   INITIATE THE CONTROLLER CHANGES
15015      GOTO 15020
15020      IF Powerflag=1 THEN
15025         ON KEY 8 LABEL "    ON" GOTO 14875
15030      ELSE
15035         ON KEY 8 LABEL "    OFF" GOTO 14875
15040      END IF
15045      ON KEY 0 LABEL Loop_label$(Lindex) GOTO 14920
15050      IF Lindex=1 AND Cindex=1 THEN
15055         ON KEY 5 LABEL Cstr$(Cindex)&"(delT)" GOTO 14905
15060      ELSE
15065         ON KEY 5 LABEL Cstr$(Cindex) GOTO 14905
15070      END IF
15075      IF Cindex>1 THEN
15080         IF Cindex=3 OR Cindex=4 THEN
15085            ON KEY 6 LABEL VAL$(Ctemp(Lindex,Cindex))&" sec." GOTO 15240
15090         ELSE
15095            ON KEY 6 LABEL VAL$(Ctemp(Lindex,Cindex)) GOTO 15240
15100         END IF
15105      ELSE
15110         IF Lindex=1 THEN
15115            ON KEY 6 LABEL VAL$(Ctemp(Lindex,Cindex))&" microV" GOTO 15240
15120         ELSE
15125            ON KEY 6 LABEL VAL$(Ctemp(Lindex,Cindex))&" degC" GOTO 15240
15130         END IF
15135      END IF
15140      ON KEY 1 LABEL " Inc.= 10↑"&VAL$(Inc_pwr) GOTO 14935
15145      C(1)=Set_temp(Lindex)
15150      FOR I=2 TO Ncpar
15155         C(I)=Cset(Lindex,I)
15160      NEXT I
15165      IF Lindex=1 THEN
15170         DISP "SP=";C(1);" microV,K=";C(2);",Ti=";C(3);",Td=";C(4);",TiBW=";C(5)*1.E+
6;",KBW=";C(6)*1.E+6;",KRF=";C(7)
15175      ELSE
15180         DISP "SP=";C(1);" C,K=";C(2);",Ti=";C(3);",Td=";C(4);",TiBW=";C(5);",KBW=";C
(6);",KRF=";C(7)
15185      END IF
15190      Tpause=TIMEDATE
15195      LOOP
15200         Tbusy=TIMEDATE-Tpause
15205         IF TIMEDATE-Tpause-Tbusy>2 THEN Tpause=TIMEDATE
15210         IF Ne<>Prev_ne THEN CALL Update_plot("N")
15215         IF Queue(1)>-1 THEN
15220            BEEP 83*23,.1
15225            SUBEXIT
15230         END IF
15235      EXIT IF TIMEDATE-Tpause>120
15240      END LOOP
15245      DISP ""
15250      SUBEXIT
15255 Manual_err:    !
15260      PRINT "   ERROR IN SUB 'MANUAL' : "&ERRM$
15265      CALL Err_record(ERRM$)
15270  SUBEND
15275 !//////////////////////////////////////////////////////////////////////
15280  SUB Chan_switch(Nchan,Dvm_cmd$)
15285 !     THIS SUB SWITCHES RELAY POSITIONS ON THE JRL SCANNER AND SETS THE
15290 !     181 DVM TO THE SPECIFIED RANGE
15295 !              Nchan : NEW JRL RELAY NUMBER
15300 !              Dvm_cmd$ : COMMAND STRING OUTPUT TO THE JLR RELAY DVM 181
15305 !      THIS SUB ASSUMES THAT THE MULTIPROGRAMMER IS SET TO THE ADDRESS 723
15310      COM /Fl/ Flag$
15315      COM /Instr/ Mh181,Jrl181,Mh195,Tap195,Bap195,Ig195
15320      COM /Jrlchan/ Chan,Tchan,Dvm_cmmd$,Default_chan
15325      ON ERROR GOTO Ch_sw_error
15330      Ch=FNChan_sig(Nchan)
15335      Write_io(723,"OP,0,255T")            ! OPEN ALL THE JRL RELAYS
15340      Ch_short=FNChan_sig(2)               ! JRL CHANNEL WITH SHORT = CHAN. 2
15345      Write_ia(723,"OP,0,"&VAL$(Ch_short)&"T")  ! CLOSE JRL RELAY IF A SHORT
15350      WAIT .1
15355      Write_ia(723,"OP,0,255T")            ! OPEN ALL THE JRL RELAYS
```

```
15360     IF Dvm_cmd$<>Dvm_cmmd$ THEN CALL Write_io(Jrl181,Dvm_cmd$)
15365     Write_io(723,"OP,0,"&VAL$(Ch)&"T")  ! CLOSE THE REQUESTED JRL RELAY
15370     Chan=Nchan
15375     Tchan=TIMEDATE
15380     SUBEXIT
15385 Ch_sw_error:    !
15390     PRINT "ERROR IN CHANNEL_SWITCH SUB"
15395     IF Flag$="OK" THEN Flag$=" CHANNEL SWITCH SUB ERROR"
15400  SUBEND
15405 !//////////////////////////////////////////////////////////////////////
15410  SUB Read_old_data
15415  !     THIS SUB READS IN THE SETUP PARAMETERS FOR A PREVIOUS RUN
15420     COM /Dt1/ File_specs$(*),Mode$(*),Gas$(*)
15425     COM /Sb3/ Fd(*),Tavg_interval
15430     ON ERROR GOTO Old_dat_err
15435     PRINT USING "@,3/"
15440     PRINT "      INPUT THE NAME OF THE FILE FROM WHICH YOU"
15445     PRINT "      WOULD LIKE TO READ A SET OF EXPERIMENTAL RUN PARAMETERS"
15450     PRINT "      TO BE USED WITH THIS RUN."
15455     PRINT
15460     PRINT "      PLACE THE DISK CONTAINING THAT FILE IN THE RIGHT DRIVE."
15465     LINPUT " ENTER FILENAME.",Fln$
15470     IF Fln$="" THEN GOTO 15465
15475     ASSIGN @I TO Fln$
15480     ENTER @I;File_specs$(*),Fd(*)
15485     ASSIGN @I TO *
15490     SUBEXIT
15495 Old_dat_err:    !
15500     PRINT USING "@,5/"
15505     PRINT "   ERROR IN OLD DATA READ ATTEMPT"
15510     PRINT "          ";ERRM$
15515     Ans$=""
15520     LINPUT "  DO YOU WANT TO TRY AGAIN? (Y/N)",Ans$
15525     IF Ans$="Y" THEN GOTO 15435
15530     SUBEXIT
15535  SUBEND
15540  !/////////////////////////////////////////////////////////////////////
15545  SUB Time_set
15550 !     THIS SUBROUTINE SETS THE SYSTEM DATE AND TIME
15555     LOOP
15560        PRINT USING "@,5/"
15565        PRINT "THE CURRENT TIME SETTING IS ",DATE$(TIMEDATE),TIME$(TIMEDATE)
15570        Ans$="N"
15575        INPUT "IS THIS TIME VALUE APPROPRIATE (Y/N)?",Ans$
15580        IF Ans$="Y" THEN SUBEXIT
15585        PRINT USING "@,10/"
15590        PRINT "INPUT THE DATE IN THE FOLLOWING FORMAT :"
15595        PRINT
15600        PRINT "            '";DATE$(TIMEDATE);"'"
15605        Str1$=DATE$(TIMEDATE)
15610        INPUT "    DAY MONTH YEAR  ",Str1$
15615        IF LEN(Str1$)<10 THEN GOTO 15585
15620        PRINT USING "@,10/,60A";"INPUT THE TIME IN 24 HOUR FORMAT :"
15625        Str2$=TIME$(TIMEDATE)
15630        INPUT "    HOURS:MINUTES  ",Str2$
15635        IF LEN(Str2$)<4 THEN GOTO 15620
15640        SET TIMEDATE DATE(Str1$)+TIME(Str2$)
15645     END LOOP
15650  SUBEND
15655  !//////////////////////////////////////////////////////////////////////
15660  SUB Plot_prep(X1,X2,Xinc,Y1,Y2,Yinc,Xtit$,Ytit$)
15665  !     THIS SUB CALCULATES THE NECESSARY GRAPHICS LIMITS
15670     COM /Mc1/ Ts,Ne,Edat(*),Pdat(*),Nr,Rtdat(*),Nf,Fedat(*),Ftdat(*),Kdat(*),Tme(*)
,Nrmax,Nfmax,Tlim
15675     COM /Gr1/ Plot_view,Plot_type,Pindex
15680     COM /Mc5/ Op_mode(*),Htemp(*),Ltemp(*),Ht_mode(*),File_num(*),Set_temp(*)
15685     COM /Mc6/ Ntm,Ntr,Ntp,Ntz
15690     COM /Sb1/ T0,Td0
15695 !      THIS SUB SETS PARAMETERS FOR THE CRT PLOTTER
15700     ALLOCATE G$[20]
15705     ON ERROR GOTO Plot_prep_error
15710     CALL G_label(G$,Plot_type,Pindex)
```

```
15715     Xtit$="TIME (MINUTES)"
15720     IF Plot_type=3 THEN
15725        Ytit$=G$&" (WATTS)"
15730     ELSE
15735        Ytit$=G$&" TEMP.(C)"
15740        IF Plot_type=2 THEN Ytit$="CORRECTED "&G$&" TEMP.(C)"
15745        IF Plot_type=1 AND Pindex=1 THEN Ytit$=G$&" EMF (microV)"
15750     END IF
15755     IF Plot_type=5 THEN Ytit$=G$&" (mW/(m*K))"
15760     !    DEFINE DEFAULT Y-AXIS VALUES FOR TEMP. BASED ON MAIN PLATE TEMP.
15765     Y1=0
15770     Y2=0
15775     Ym=Set_temp(5)
15780     Yinc=Ym/5
15785     Yinc=INT(Yinc/10)*10
15790     IF Yinc=0 THEN Yinc=10
15795     REPEAT
15800        Y2=Y2+Yinc
15805     UNTIL Y2>Yinc+Ym
15810     SELECT Plot_type
15815     CASE 1        !  FULL SCALE Y_AXIS DEFAULT VALUES FOR GAP tc
15820        IF Pindex=1 THEN
15825           Y1=-500
15830           Y2=500
15835           Yinc=100
15840        END IF
15845     CASE 2        !  FULL SCALE Y_AXIS DEFAULT VALUES FOR tc TEMPS.
15850        Y1=Y2-5*Yinc
15855     CASE 3        !  FULL SCALE Y-AXIS DEFAULT VALUES FOR MAIN HEATER POWER
15860        Y2=100
15865        Yinc=10
15870     CASE 4        !  FULL SCALE Y-AXIS DEFAULT VALUES FOR REF. BLOCK
15875        Y1=18
15880        Y2=30
15885        Yinc=2
15890     CASE 5    ! FULL SCALE Y-AXIS DEFAULT VALUES FOR THERM. COND. (mW/M*K)
15895        Y1=0
15900        Y2=100
15905        Yinc=10
15910     CASE ELSE
15915     END SELECT
15920     Xrange=1800  !  30 MIN. RANGE
15925     X1=0
15930     Xinc=300
15935     Fact=INT(((Ts*Ne)/Xrange)+1)
15940     IF (Fact*Xrange-Ts*Ne)<.2*Xrange THEN Fact=Fact+1
15945     X2=Xrange*Fact
15950     IF Plot_type=2 OR Plot_type=5 THEN
15955        X1pt2=INT(Td0/Xinc)*Xinc
15960        X1=X1pt2
15965        Fact=(INT((Ts*Ne-X1pt2)/Xrange)+1)
15970        IF (Fact*Xrange-Ts*Ne+Td0)<.2*Xrange THEN Fact=Fact+1
15975        X2=X1+Fact*Xrange
15980     END IF
15985     IF Plot_view>0 OR Nf>2 THEN      !   X-AXIS MODIFY & Y-AXIS AUTO SCALE
15990        IF Plot_view=0 THEN
15995           Xinc=Xinc*Fact
16000        ELSE
16005           Xrange=Xrange*(2↑(Plot_view-1))  !   ADJUST XRANGE FOR PLOT_VIEW
16010           Xinc=Xinc*(2↑(Plot_view-1))      !   ADJUST XINC   FOR PLOT_VIEW
16015           X1=X2-Xrange
16020           IF X1<0 THEN
16025              X1=0
16030              X2=X1+Xrange
16035           END IF
16040           IF (Plot_type=2 OR Plot_type=5) AND X1<X1pt2 THEN
16045              X1=X1pt2
16050              X2=X1+Xrange
16055           END IF
16060           WHILE Ne*Ts<X1+.8*Xrange   ! SHOW 80% OF THE DATA IN LAST WINDOW
16065              X1=X1-Xinc
16070           END WHILE
```

```
16075          IF X1<=0 THEN X1=0
16080          X2=X1+Xronge
16085       END IF
16090 !             SCALE Y AXIS
16095       Nes=INT(X1/Ts)
16100       IF Nes>Ne THEN Nes=Ne
16105       IF Nes<=0 THEN Nes=1
16110       Nstp=1
16115       WHILE (X1>Tme(Nstp)+30) AND Nstp<Nf
16120          Nstp=Nstp+1
16125       END WHILE
16130       SELECT Plot_type
16135       CASE 1
16140          Top=Ne-Nes+1
16145          ALLOCATE Arr(1:Top)
16150          FOR J=1 TO Top
16155             Arr(J)=Edot(Pindex,Nes+J-1)
16160          NEXT J
16165          Stp=MIN(Arr(*))
16170          Endp=MAX(Arr(*))
16175          IF Pindex=1 THEN
16180             Stp=Stp*1.E+6        !  MICROVOLT CONVERSION
16185             Endp=Endp*1.E+6
16190          END IF
16195          Tpstep=.001             !  MIN. DIFF. ACCEPTED FOR RTD TEMPS. (C)
16200          IF Pindex=1 THEN Tpstep=.1          !  MIN. STEP FOR MICROVOLTS
16205       CASE 2
16210          Top=Nf-Nstp+1
16215          IF Kdot(Nf)=0 THEN Top=Top-1
16220          IF Top<1 THEN Top=1
16225          ALLOCATE Arr(1:Top)
16230          FOR J=1 TO Top
16235             Arr(J)=Ftdat(1+2*(Pindex-1),Nstp+J-1)+Ftdot(2+2*(Pindex-1),Nstp+J-1)
16240          NEXT J
16245          Stp=MIN(Arr(*))
16250          Endp=MAX(Arr(*))
16255          Tpstep=.0001 !  MIN. DIFF. ACCEPTED FOR T.C. TEMPS. (C)
16260       CASE 3
16265          Top=Ne-Nes+1
16270          ALLOCATE Arr(1:Top)
16275          FOR J=1 TO Top
16280             Arr(J)=Pdot(1,Nes+J-1)*Pdot(2,Nes+J-1)
16285          NEXT J
16290          Stp=MIN(Arr(*))
16295          Endp=MAX(Arr(*))
16300          Tpstep=.001   !  MIN. DIFF. ACCEPTED FOR POWER (WATTS)
16305       CASE 4
16310          ALLOCATE Arr(1:Nr)
16315          FOR J=1 TO Nr
16320             Arr(J)=Rtdat(1,J)
16325          NEXT J
16330          Stp=MIN(Arr(*))
16335          Endp=MAX(Arr(*))
16340          Tpstep=.001   !  MIN. DIFF. ACCEPTED FOR REF. TEMP. PLOT (Y AXIS)
16345       CASE 5
16350          Top=Nf-Nstp+1
16355          IF Kdat(Nf)=0 THEN Top=Top-1
16360          IF Top<1 THEN Top=1
16365          ALLOCATE Arr(1:Top)
16370          FOR J=1 TO Top
16375             Arr(J)=Kdot(Nstp+J-1)
16380          NEXT J
16385          Stp=MIN(Arr(*))
16390          Endp=MAX(Arr(*))
16395          Tpstep=.001    ! MIN. DIFF. ACCEPTED FOR K VALUE (Y AXIS)
16400       CASE ELSE
16405       END SELECT
16410       IF Stp>Endp THEN
16415          Lo=Endp
16420          Hi=Stp
16425       ELSE
16430          Lo=Stp
```

```
16435          Hi=Endp
16440       END IF
16445       Pstep=ABS(Stp-Endp)
16450       IF Pstep=0 OR Pstep<Tpstep THEN Pstep=Tpstep
16455       DEALLOCATE Arr(*)
16460       IF Pstep>1 THEN
16465          Ex=0
16470          WHILE Pstep>=10
16475             Ex=Ex+1
16480             Pstep=Pstep/10
16485          END WHILE
16490       ELSE
16495          Ex=0
16500          WHILE Pstep<1
16505             Ex=Ex-1
16510             Pstep=Pstep*10
16515          END WHILE
16520       END IF
16525       Yinc=10↑Ex
16530       Y3fudge=1
16535       IF Pstep<5 THEN
16540          Yinc=Yinc/2
16545          Y3fudge=2*Y3fudge
16550       END IF
16555       Y1=INT(Lo/Yinc)*Yinc-Yinc
16560       Y2=INT(Hi/Yinc+1)*Yinc+Yinc
16565    ELSE                    !  FULL TIME VIEW PLOT
16570       Xinc=Xinc*Fact
16575    END IF
16580    SUBEXIT
16585 Plot_prep_error:   !
16590    PRINT "PLOT PREP ERROR  ";ERRM$
16595    CALL Err_record(ERRM$)
16600  SUBEND
16605 !/////////////////////////////////////////////////////////////////////////
16610  SUB Update_plot(Rsp$)
16615  !        THIS SUB PLOTS THE MOST RECENT DATA POINT ON THE CRT PLOT
16620  !        CURRENTLY BEING DISPLAYED.
16625  !        INPUT VARS.
16630  !              Rsp$ = FLAG FOR PLOT RESCALE  ('Y' ALLOWS A PLOT RESCALE,
16635  !                     IF NECESSARY; 'N' DOES NOT PERMIT A PLOT RESCALE)
16640    COM /Flgs/ Igflag,Prev_ne
16645    COM /Gr1/ Plot_view,Plot_type,Pindex
16650    COM /Gr2/ X1,X2,Xinc,Y1,Y2,Yinc,Xtit$[40],Ytit$[40]
16655    COM /Mc1/ Ts,Ne,Edat(*),Pdat(*),Nr,Rtdat(*),Nf,Fedat(*),Ftdat(*),Kdat(*),Tme(*)
,Nrmax,Nfmax,Tlim
16660    COM /Tcst1/ Stare_flag
16665    ON ERROR GOTO Upd_err
16670    IF Ne<1 THEN SUBEXIT
16675 !         UPDATE CURRENT SCREEN PLOT
16680    Prev_ne=Ne
16685    SELECT Plot_type
16690    CASE 1
16695       Ypl=Edat(Pindex,Ne)
16700       IF Pindex=1 THEN Ypl=Ypl*1.E+6
16705       Xpl=Ne*Ts
16710    CASE 2
16715       IF Nf<>0 THEN
16720          IF Kdat(Nf)<>0 THEN
16725             Ypl=Ftdat(1+2*(Pindex-1),Nf)+Ftdat(2+2*(Pindex-1),Nf)
16730             Xpl=Tme(Nf)
16735          ELSE
16740             SUBEXIT
16745          END IF
16750       END IF
16755    CASE 3
16760       Ypl=Pdat(1,Ne)*Pdat(2,Ne)
16765       Xpl=Ne*Ts
16770    CASE 4
16775       Ypl=Rtdat(1,Nr)
16780       Xpl=Rtdat(2,Nr)
16785    CASE 5
```

```
16790         IF Nf<>0 THEN
16795            IF Kdat(Nf)<>0 THEN
16800               Ypl=Kdat(Nf)
16805               Xpl=Tme(Nf)
16810            ELSE
16815               SUBEXIT
16820            END IF
16825         END IF
16830      CASE ELSE
16835      END SELECT
16840      IF Rsp$="Y" AND (Ypl>Y2 OR Ypl<Y1 OR Xpl>(X1+(X2-X1)*1.0)) AND Store_flag=0 THE
N CALL Rescale_plot(Xpl,Ypl)
16845      IF (Plot_type=2 OR Plot_type=5) THEN
16850         IF Nf=1 THEN MOVE Xpl,Ypl
16855         IF Nf>1 THEN
16860            IF Tme(Nf)<>0 THEN PLOT Xpl,Ypl
16865         END IF
16870      ELSE
16875         PLOT Xpl,Ypl
16880      END IF
16885      SUBEXIT
16890 Upd_err:    !
16895      PRINT " PLOTTING ERROR IN SUB ' UPDATE_PLOT' : "&ERRM$
16900      CALL Err_record(ERRM$)
16905   SUBEND
16910 !////////////////////////////////////////////////////////////////////////
16915   SUB Atod_io(V,Tread,Oldvolt)
16920    !    THIS SUB TAKES AND RETURNS AN AVERAGED A/D READING. THE VOLTAGE
16925    !    READING RETURNED FROM THIS SUB IS THE OUTER GUARD RTD VOLTAGE.
16930      COM /Read1/ Io_error,Bad_instr(*),Bad_read_time$(*)
16935      COM /Sb1/ T0,Td0
16940      COM /Fl/ Flag$
16945      ON TIMEOUT 7,.1 GOTO Read_it_error
16950      ON ERROR GOTO Read_it_error2
16955      Vt=0
16960      FOR I=1 TO 20
16965         OUTPUT 723;"IP,2T"
16970         ENTER 72301;V
16975         Vt=Vt+V
16980      NEXT I
16985      V=Vt/20                 !    CALCULATE AVERAGE VALUE
16990      V=V/10                  !    SCALE THE MULTIPROGRAMMER OUTPUT
16995      Tread=TIMEDATE-T0
17000      OFF TIMEOUT
17005      SUBEXIT
17010 Read_it_error:   !
17015      Io_error=Io_error+1
17020      Bad_instr(Io_error)=723
17025      CLEAR 723
17030      Bad_read_time$(Io_error)=DATE$(TIMEDATE)&" "&TIME$(TIMEDATE)
17035      BEEP 83*35,.1
17040      IF Io_error=100 THEN Flag$="MAXIMUM NUMBER OF IO ERRORS EXCEEDED !!"
17045      V=Oldvolt
17050      Tread=TIMEDATE-T0
17055 Read_it_error2: !
17060      CALL Err_record(ERRM$)
17065      PRINT " ERROR IN 'Atod_io' SUB : "&ERRM$
17070      BEEP 83*35,.1
17075   SUBEND
17080   !//////////////////////////////////////////////////////////////////////
17085   SUB Data_read
17090   !     THIS SUB TAKES JRL CHANNEL NUMBERS FROM THE Queue(*) ARRAY,
17095   !     INCREMENTS THE Queue(*) ARRAY, THEN DIRECTS THE JRL RELAY TO
17100   !     SWITCH TO THAT CHANNEL. AFTER A DESIGNATED WAITING PERIOD, A
17105   !     VOLTAGE READING IS TAKEN AND STORED IN THE APPROPRIATE ARRAY
17110   !     STORAGE LOCATION.
17115   !  VARIABLES :
17120   !       Settle_time = TIME (SEC) BETWEEN THE CHANNEL SWITCH AND READING
17125   !
17130      COM /Adjloc2/ Last_reading(*)
17135      COM /Dr1/ Bad_curr
17140      COM /Dr2/ Rtdi_hist(1:10)
```

```
17145     COM /Fld/ Disp_flag
17150     COM /Ioscan/ Queue(*),Nq,Qseq1(*),Nqs1,Qseq2(*),Nqs2,Qseq3(*),Nqs3,Qseq4(*),Nqs
4,Qseq5(*),Nqs5,Qseq6(*),Nqs6
17155     COM /Htr2/ New_htrcur
17160     COM /Flgs/ Igflag,Prev_ne
17165     COM /Instr/ Mh181,Jrl181,Mh195,Tap195,Bap195,Ig195
17170     COM /Jrlchan/ Chan,Tchan,Dvm_cmmd$,Default_chan
17175     COM /Mc1/ Ts,Ne,Edat(*),Pdat(*),Nr,Rtdat(*),Nf,Fedat(*),Ftdat(*),Kdat(*),Tme(*)
,Nrmax,Nfmax,Tlim
17180     COM /Mc2/ Heater_sres
17185     COM /Mc3/ Rtdpwr_sres
17190     COM /Rn/ Run
17195     COM /Sb1/ T0,Td0
17200     COM /Sb2/ I_rtd,Tref,Emf_ref
17205     COM /Sb3/ Fd(*),Tavg_interval
17210     COM /Sdisp/ Screen_prnt
17215     COM /Water/ Ncw,Cwater(*)
17220     COM /Zeros/ Zjrl181_200,Zjrl181_20,Zgap195,Zhistory(*)
17225     ON KBD GOSUB Nothing2
17230     ON KEY 0 LABEL "" GOSUB Nothing2
17235     ON KEY 5 LABEL "" GOSUB Nothing2
17240     ON ERROR GOTO Data_read_err
17245     DISP "           JRL DVM READING IN PROGRESS ... DON'T TOUCH ANY KEYS!!"
17250     Settle_time=5                  !  DVM SETTLING TIME (SECONDS)
17255     Disp_flag=TIMEDATE+Settle_time-9
17260     Nchan=Queue(1)
17265     IF (Nchan>2 AND Nchan<7) OR (Nchan>12 AND Nchan<17) THEN
17270        CALL Chan_switch(Nchan,"R2B1X")
17275     ELSE
17280        CALL Chan_switch(Nchan,"R3B1X")
17285     END IF
17290  !  UPDATE THE QUEUE
17295     FOR I=1 TO Nq-1
17300        Queue(I)=Queue(I+1)
17305     NEXT I
17310     Queue(Nq)=-1
17315     REPEAT
17320        IF Ne<>Prev_ne THEN
17325           CALL Update_plot("N")
17330           Tinterupt=TIMEDATE
17335        END IF
17340     UNTIL TIMEDATE-Tchan>Settle_time
17345     Vlast=Last_reading(Chan)
17350     IF Chan=19 THEN
17355        SYSTEM PRIORITY 15
17360        WAIT 2-(TIMEDATE-Tinterupt)
17365     END IF
17370     IF (Nchan>2 AND Nchan<7) OR (Nchan>12 AND Nchan<17) THEN
17375        CALL Read_io(V,Tmr,Jrl181,"R2B1X",Vlast)
17380     ELSE
17385        CALL Read_io(V,Tmr,Jrl181,"R3B1X",Vlast)
17390     END IF
17395     IF Chan=8 THEN                  !  TO ZERO GAP DVM TAKE SIMULTANEOUS READ
17400        CALL Read_io(Vg,Tt,Ig195,"R0X",Last_reading(Chan))
17405     END IF
17410     Last_reading(Chan)=V
17415     Chan0=Chan
17420  !        STORE THE 181 READING IN THE APPROPRIATE PLACE
17425     SELECT Chan0
17430     CASE 1
17435        CALL Ref_rtd(V,Tmr)
17440     CASE 2         !      CORRECT JRL DVM ZEROS
17445        Zhistory(1,2)=Zhistory(1,1)
17450        Zjrl181_200=Zjrl181_200+V
17455        Zhistory(1,1)=Zjrl181_200
17460        CALL Chan_switch(Nchan,"R2B1X")
17465        REPEAT
17470        UNTIL TIMEDATE-Tchan>Settle_time
17475        CALL Read_io(V,Tmr,Jrl181,"R2B1X",0)
17480        Zhistory(2,2)=Zhistory(2,1)
17485        Zjrl181_20=Zjrl181_20+V
17490        Zhistory(2,1)=Zjrl181_20
```

```
17495     CASE 18
17500        Laxcurr=.00001/(1+INT((TIMEDATE-T0)/(60*50)))
17505        IF TIMEDATE-T0<60*50 THEN Laxcurr=.0001
17510        IF Rtdi_hist(1)=0 THEN Rtdi_hist(1)=I_rtd
17515        IF (ABS(Rtdi_hist(1)-V/Rtdpwr_sres)<.0000001) OR (TIMEDATE-T0<60*150 AND ABS
(I_rtd-V/Rtdpwr_sres)<Laxcurr) THEN
17520           FOR Jay=9 TO 1 STEP -1
17525              Rtdi_hist(Jay+1)=Rtdi_hist(Jay)
17530           NEXT Jay
17535           Rtdi_hist(1)=V/Rtdpwr_sres
17540           IF ((TIMEDATE-T0>60*120) OR Run>1) AND Rtdi_hist(10)<>0 THEN
17545              I_rtd=(SUM(Rtdi_hist))/10
17550           ELSE
17555              I_rtd=Rtdi_hist(1)
17560           END IF
17565           IF Screen_prnt THEN PRINT "   RTD CURRENT = ";I_rtd
17570        ELSE
17575           Bad_curr=Bad_curr+1
17580           PRINT "  BAD RTD CURRENT READING!  V/R=";V/Rtdpwr_sres;" ,  OLD I_rtd=";I
_rtd
17585           CALL Err_record("  BAD RTD CURRENT READING!  V/R="&VAL$(V/Rtdpwr_sres)&"
,  OLD I_rtd="&VAL$(I_rtd))
17590           IF Screen_prnt THEN OUTPUT 701;"  BAD RTD CURRENT READING! I_RTD=";I_rtd;
" , V/R=";V/Rtdpwr_sres
17595           BEEP 83,.2
17600           IF Bad_curr>3 THEN
17605              IF V/Rtdpwr_sres>.002 OR V/Rtdpwr_sres<.0001 THEN
17610                 Flag$="  RTD CURRENT POWER SUPPLY PROBLEM (FLUKE)"
17615              ELSE
17620                 FOR Jay=9 TO 1 STEP -1
17625                    Rtdi_hist(Jay+1)=Rtdi_hist(Jay)
17630                 NEXT Jay
17635                 Rtdi_hist(1)=V/Rtdpwr_sres
17640                 I_rtd=Rtdi_hist(1)
17645                 IF Rtdi_hist(10)<>0 THEN I_rtd=SUM(Rtdi_hist)/10
17650                 Bad_curr=0
17655              END IF
17660           END IF
17665        END IF
17670     CASE 19
17675        Pdat(1,Ne)=V/Heater_sres
17680        New_htrcur=1
17685     CASE 3 TO 6,13 TO 16
17690        IF Td0=0 THEN
17695           CALL Rtd_tune(Chan0,V)
17700        ELSE
17705           CALL Tc_store(Chan0,V,Tmr)
17710        END IF
17715     CASE 7
17720        Fd(20)=V                             !  TOP IG
17725     CASE 8
17730        IF Igflag=1 THEN
17735           Fd(19)=V                          !  TOTAL IG
17740           Zhistory(3,2)=Zhistory(3,1)
17745           Zhistory(3,1)=Zgap195+(Vg-V)
17750        ELSE
17755           Zhistory(3,2)=Zhistory(3,1)
17760           Zgap195=Zgap195+(Vg-V)            !  CORRECT GAP DVM ZERO
17765           Zhistory(3,1)=Zgap195
17770        END IF
17775     CASE 9
17780        Fd(18)=V                             !  BOTTOM IG
17785     CASE 11
17790        Ncw=Ncw+1
17795        Cwater(1,Ncw)=(INT((FNTemp_tc(V))*1000.))/1000.
17800        Cwater(2,Ncw)=1.0*INT(Tmr/60)
17805     CASE ELSE
17810     END SELECT
17815     SUBEXIT
17820 Nothing2:     !
17825     BEEP 83*23,.1
17830     RETURN
```

```
17835 Data_read_err:     !
17840     PRINT "  ERROR IN THE SUB 'DATA_READ'"
17845     PRINT ERRM$
17850     CALL Err_record(ERRM$)
17855     BEEP 83*25,.1
17860  SUBEND
17865  !///////////////////////////////////////////////////////////////////////
17870  SUB K_stor
17875  !    THIS SUB COMPUTES THE RUNTIME VALUES OF T.C. TEMPERATURES (PHASE 2)
17880  !    AND THERMAL CONDUCTIVITY USED IN PLOTTING AND STAT. ANALYSIS.
17885     COM /Mc1/ Ts,Ne,Edat(*),Pdat(*),Nr,Rtdat(*),Nf,Fedat(*),Ftdat(*),Kdat(*),Tme(*)
,Nrmax,Nfmax,Tlim
17890     COM /Sb2/ I_rtd,Tref,Emf_ref
17895     COM /Sb3/ Fd(*),Tavg_interval
17900     ON ERROR GOTO K_stor_err
17905     !           RECOMPUTE THE PLATE TEMP. USING AVG. REF. BLOCK TEMP.
17910     Avg_tref=(Rtdat(1,Nr)+Rtdat(1,Nr-1))/2
17915     Emf_ref=FNEmf_tc(Avg_tref,1)
17920     FOR I=1 TO 7 STEP 2
17925        Ftdat(I,Nf)=FNTemp_tc(Fedat(I,Nf))
17930        Ftdat(I+1,Nf)=FNTemp_tc(Fedat(I,Nf)-Fedat(I+1,Nf)/9)-Ftdat(I,Nf)
17935     !      NOTE: THE Delta T (STORED IN Ftdat) IS COMPUTED ABOVE BASED ON
17940     !                   1) 9 ARMS OF THE STAR THERMOCOUPLE
17945     !                   2) A POSITIVE STAR tc SIGNAL CORRESPONDS TO
17950     !                        PLATE CENTER BEING WARMER THAN THE EDGE.
17955     !                   3) Delta T IS DEFINED AS THE DIFFERENCE :
17960     !                        AVERAGE PLATE TEMP.-CENTER PLATE TEMP.
17965     NEXT I
17970     !           ASSIGN VARIABLES USED IN THE SUB CALL FOR k CALC.
17975     Ntme=INT(Tme(Nf)/Ts)
17980     Q=Pdat(1,Ntme)*Pdat(2,Ntme)*1000   !  CONVERTED TO MILLIWATTS
17985     T1=Ftdat(7,Nf)+Ftdat(8,Nf)         !  B.A.P. TEMP. (CORRECTED)
17990     T2=Ftdat(5,Nf)+Ftdat(6,Nf)         !  B.M.P. TEMP. (CORRECTED)
17995     T3=Ftdat(3,Nf)+Ftdat(4,Nf)         !  T.M.P. TEMP. (CORRECTED)
18000     T4=Ftdat(1,Nf)+Ftdat(2,Nf)         !  T.A.P. TEMP. (CORRECTED)
18005     Dia=Fd(8)*100.
18010     Dx=Fd(3)*100.
18015     Dr=Fd(21)*100.
18020     Sc=Fd(22)                          !  PLATE SPACER CODE
18025     Rc=Fd(15)     ! RUN MODE (DBLE SIDED, TOP, BOT.)
18030     CALL K_ghp(K,Tlo,Thi,Dxc,Acor,T1,T2,T3,T4,Q,Dia,Dx,Dr,Sc,Rc,0)
18035     Kdat(Nf)=K                         !  LOAD ARRAY IN   mW/(m*K)   UNITS
18040     SUBEXIT
18045 K_stor_err:    !
18050     PRINT "    ERROR IN 'K_STOR' SUB"
18055     CALL Err_record(ERRM$)
18060     BEEP 83*3,.1
18065  SUBEND
18070  !/////////////////////////////////////////////////////////////////
18075  SUB Run_abort
18080  !    THIS SUB SETS THE RUN ABORT VARIABLES
18085     COM /Fl/ Flag$
18090     CALL Poweroff
18095     Flag$="RUN ABORTED"
18100  SUBEND
18105  !/////////////////////////////////////////////////////////////////
18110  SUB Rtd_tune(Ch,V)
18115   !      THIS SUBROUTINE ADJUSTS A SETPOINT OFFSET THAT CALIBRATES
18120   !      THE RTD TEMPERATURE SETTING TO THE tc READING FOR EACH
18125   !      PLATE.
18130     COM /Conv_dat/ R0(*),Alpha(*),Delta(*)
18135     COM /Mc1/ Ts,Ne,Edat(*),Pdat(*),Nr,Rtdat(*),Nf,Fedat(*),Ftdat(*),Kdat(*),Tme(*)
,Nrmax,Nfmax,Tlim
18140     COM /Mc5/ Op_mode(*),Htemp(*),Ltemp(*),Ht_mode(*),File_num(*),Set_temp(*)
18145     COM /Rtd_corr/ Tcorr_rtd,Rtd_adj_flag,Sp_corr(*)
18150     COM /Sdisp/ Screen_prnt
18155     COM /Tune1/ Atune(*),Ok_flag(*),Splast(*)
18160     ON ERROR GOTO Tune_err
18165     IF Ch<10 THEN
18170        Atune(Ch-2)=V
18175     ELSE
18180        Atune(Ch-12)=FNTemp_tc(Atune(Ch-12)-V/9)
```

```
18185         IF Atune(Ch-12)<0 THEN SUBEXIT
18190         SELECT Ch
18195         CASE 15
18200            Avg=(Atune(2)+Atune(3))/2
18205            Trtd=Edot(5,Ne-2)
18210            Splost(5,2)=Splost(5,1)
18215            Splost(5,1)=Avg-Trtd
18220            Sp_corr(5)=Splost(5,1)-(Splost(5,1)-Sp_corr(5))*.5
18225            CALL Set_pnt_colc
18230            IF ABS(Sp_corr(5)-Splost(5,1))<.01 AND ABS(Set_temp(5)-Avg)<.05 THEN Ok_f
log(3)=1
18235            IF Screen_prnt THEN
18240               T1=INT(Avg*10000.)/10000.
18245               T2=INT(Trtd*10000.)/10000.
18250               Tdel=INT(Sp_corr(5)*10000.)/10000.
18255               OUTPUT 701;"   MAIN PLATE : T.C.=";T1;" ,  RTD=";T2;" ,    T diff. (T.C
.-RTD) = ";T1-T2
18260               New_sp=INT((Set_temp(5)-Sp_corr(5))*10000.)/10000.
18265               OUTPUT 701;"        NEW SET POINT = ";New_sp;" ,    SET POINT CORRECTIO
N = ";Tdel
18270            END IF
18275         CASE 13
18280            Trtd=Edat(3,Ne-1)
18285            Splost(3,2)=Splost(3,1)
18290            Splost(3,1)=Atune(1)-Trtd
18295            Sp_corr(3)=Splost(3,1)-(Splost(3,1)-Sp_corr(3))*.5
18300            CALL Set_pnt_colc
18305            IF ABS(Sp_corr(3)-Splost(3,1))<.01 AND ABS(Set_temp(3)-Atune(1))<.05 THEN
 Ok_flog(1)=1
18310            IF Screen_prnt THEN
18315               T1=INT(Atune(1)*10000.)/10000.
18320               T2=INT(Trtd*10000.)/10000.
18325               Tdel=INT(Sp_corr(3)*10000.)/10000.
18330               OUTPUT 701;"   TOP PLATE : T.C.=";T1;" ,  RTD=";T2;" ,    T diff. (T.C.
-RTD) = ";T1-T2
18335               New_sp=INT((Set_temp(3)-Sp_corr(3))*10000.)/10000.
18340               OUTPUT 701;"        NEW SET POINT = ";New_sp;" ,    SET POINT CORRECTIO
N = ";Tdel
18345            END IF
18350         CASE 16
18355            Trtd=Edat(4,Ne-1)
18360            Splost(4,2)=Splost(4,1)
18365            Splost(4,1)=Atune(4)-Trtd
18370            Sp_corr(4)=Splost(4,1)-(Splast(4,1)-Sp_corr(4))*.5
18375            CALL Set_pnt_colc
18380            IF ABS(Sp_corr(4)-Splost(4,1))<.01 AND ABS(Set_temp(4)-Atune(4))<.05 THEN
 Ok_flog(4)=1
18385            IF Screen_prnt THEN
18390               T1=INT(Atune(4)*10000.)/10000.
18395               T2=INT(Trtd*10000.)/10000.
18400               Tdel=INT(Sp_corr(4)*10000.)/10000.
18405               OUTPUT 701;" BOTTOM PLATE : T.C.=";T1;" ,  RTD=";T2;" ,    T diff. (T.C
.-RTD) = ";T1-T2
18410               New_sp=INT((Set_temp(4)-Sp_corr(4))*10000.)/10000.
18415               OUTPUT 701;"        NEW SET POINT = ";New_sp;" ,    SET POINT CORRECTIO
N = ";Tdel
18420            END IF
18425            IF SUM(Ok_flag)=3 THEN
18430               Rtd_adj_flag=1
18435               BEEP 83*3,.2
18440               OUTPUT 701;"           ALL SETPOINT OFFSETS ARE STABILIZED TO .01 K"
18445            ELSE
18450               MAT Ok_flag= (0)
18455            END IF
18460         CASE ELSE
18465         END SELECT
18470      END IF
18475      SUBEXIT
18480 Tune_err:    !
18485      PRINT "   ERROR IN THE 'RTD_TUNE' SUB : "&ERRM$
18490      CALL Err_record(ERRM$)
18495   SUBEND
```

```
18500  !//////////////////////////////////////////////////////////////////////
18505  SUB Err_record(M$)
18510  !   THIS SUB RECORDS ALL NON-I.O. ERRORS IN THE STRING ARRAY 'RUN_ERRORS$'
18515     COM /Fl/ Flag$
18520     COM /Run_err/ Rterr,Run_errors$(*),Err_max
18525     ON ERROR GOTO Err_err
18530     Rterr=Rterr+1
18535     IF Rterr>=Err_max THEN
18540        Flag$="   THE MAXIMUM NUMBER OF PROGRAM ERRORS HAS BEEN EXCEEDED!!"
18545     ELSE
18550        Run_errors$(Rterr)=M$
18555     END IF
18560     SUBEXIT
18565 Err_err:    !
18570     Flag$="  ERROR IN 'Err_record' SUB "&ERRM$
18575  SUBEND
```

Appendix B: A Thermocouple Device for Determination of Average Surface Temperature*

J. G. Hust and David R. Smith
Chemical Engineering Science Division
National Bureau of Standards
Boulder, Colorado

Abstract

A thermocouple-based device for the measurement of average surface temperature is described. The device requires the measurement of only two emfs and yields the average temperature over the entire surface instrumented with the device. It consists of a single (normal) thermocouple and a thermopile-like element which performs the averaging. A particular use of the device is described to illustrate its utility.

Key words: average temperature; thermocouple; thermopile; thermal properties.

INTRODUCTION

The determination of thermal properties of materials frequently requires the measurement of the average temperature of a surface. This is usually done by placing more than one temperature sensor on the surface and averaging the resultant temperatures. The number of such sensors is usually small to simplify wiring and reduce cost and complexity of instrumentation. As a result, the average temperature obtained is often based on an inadequate sampling of the surface. Depending on the degree of temperature nonuniformity, the error in such measurements may be a significant contribution to the total error involved in thermal property determinations. For example, the determination of surface temperatures for thermal conductivity measurements using guarded hot plate or heat flow meter apparatus can be significantly affected by such errors.

The device described here has only three leads but is capable of providing a much more accurate average temperature of the surface than that obtained with two thermocouples.

PHYSICAL DESCRIPTION

The device is illustrated in Figure 1. It consists of a normal thermocouple of alloys A and B that measures the temperature of point 0 of the surface. At that point, a third wire consisting of a series of alternating thermocouple elements (A,B,A...B) (similar to a thermopile) is attached. The points of connection between the alternating thermocouple elements are labeled 1,2,3,...

In this example, we have chosen 18 points for illustration, but the number can be any even number. The wire exits the surface from the last point, forming with the original thermocouple a set of three measurement leads. From wires A_1 and B one obtains the temperature of point O. From wires A_1 and A_2 one obtains a measure of the average difference in temperature between the even-numbered points and the odd-numbered points. By combining the two results one obtains a measure of the average temperature of the surface.

MATHEMATICAL DESCRIPTION

The emf, $e(T_r, T_o)$, produced by a single thermocouple at temperature, T_o, with respect to a reference temperature, T_r, can be expressed by eq. (1)

$$e(T_r,T_o) = \int_{T_r}^{T_O} S_A\, dT + \int_{T_O}^{T_r} S_B dT = \int_{T_r}^{T_O} S_{AB} dT \tag{1}$$

where S_A is the thermopower of wire A, S_B is the thermopower of wire B, and $S_{AB} = S_A - S_B$. See reference [1]. The emf e_p produced by the thermopile-like elements can thus be written as

$$e_p(T_o,T_{18}) = \int_{T_O}^{T_1} S_A dT + \int_{T_1}^{T_2} S_B dT + \ldots\ldots + \int_{T_{16}}^{T_{17}} S_A dT + \int_{T_{17}}^{T_{18}} S_B dT \tag{2}$$

where it is assumed that $T_o = T_{18}$ because of their physical proximity, and thus the emf from the wires A_1 and A_2 (of the same material) between the surface and reference temperature is zero. If, in addition, we assume that the thermopowers are nearly constant over the temperature range of the surface, we can approximate

$$e_p\ (T_O,T_{18}) = S_A(T_1-T_O)+S_B(T_2-T_1)+\ldots+S_A(T_{17}-T_{16})+S_B(T_{18}-T_{17}) \tag{3}$$

This, using $T_O=T_{18}$, yields

$$e_p(T_o,T_{18}) = S_{AB}\ \left(\sum_{j=1}^{9} T_{2j-1} - \sum_{j=1}^{9} T_{2j} \right) \tag{4}$$

This can be rewritten as

$$e_p(T_0, T_{18})/(9\ S_{AB}) = \bar{T}_{odd} - \bar{T}_{even} \tag{5}$$

yielding the average temperature difference of the odd-numbered points with respect to the even-numbered points. If the even-numbered points are placed in proximity to each other, we can assume that their average temperature is the same as T_0 and can write

$$\bar{T} = T_0 + e_p(T_0,T_{18})/(9\ S_{AB}), \tag{6}$$

where $\bar{T}$ is the average temperature of the surface as indicated by the odd-numbered points.

EXPERIMENTAL VERIFICATION

A high-temperature guarded-hot-plate thermal conductivity apparatus that uses this device was recently completed at the National Bureau of Standards, Boulder, Colorado [2]. The utility of this device is clearly demonstrated from an analysis of surface temperature difference measurements used in the calculation of thermal conductivity near ambient temperature as a function of axial temperature difference across the specimens. Measurements were performed both in the double-sided and single-sided mode of operation [2]. The axial temperature differences through the specimens ranged from those typically used in a guarded-hot-plate apparatus, approximately 20°C, to very small differences, 1°C. The purpose of these measurements is to detect the existence of systematic errors in the determination of the axial temperature difference through the specimens. The presence of such systematic errors, if they are not linearly dependent on axial temperature difference, will be exhibited by an increasingly larger error in thermal conductivity as the axial temperature difference approaches zero.

Figure 2 shows the results of our double-sided mode [2] thermal conductivity measurements on a pair of insulation specimens. In addition, we have included the results that were calculated from the temperatures of the plates as determined by the single thermocouple at point 0. As can be seen, the temperature-averaging device produces thermal conductivities that are much less dependent on axial temperature difference. This indicates that the measurement technique using a single thermocouple has a significantly larger bias due to radial temperature variations.

Figures 3 and 4 show the same type of results for the single-sided mode of operation in which the top and bottom specimens were measured separately. Again the thermal conductivity results are less dependent on axial temperature difference when average surface temperatures are used.

REFERENCES:

[1] Manual on the Use of Thermocouples in Temperature Measurement, ASTM STP 470B, (ASTM, Philadelphia, PA, 1981) p.4

[2] Hust, J. G.; Filla, B. James; Hurley, J. A. and Smith, David R.; An Automated High Temperature Guarded-Hot-Plate Apparatus for Measuring Thermal Conductivity of Insulation Between 300 and 800 K, to be published as an NBSIR.

List of Figures.

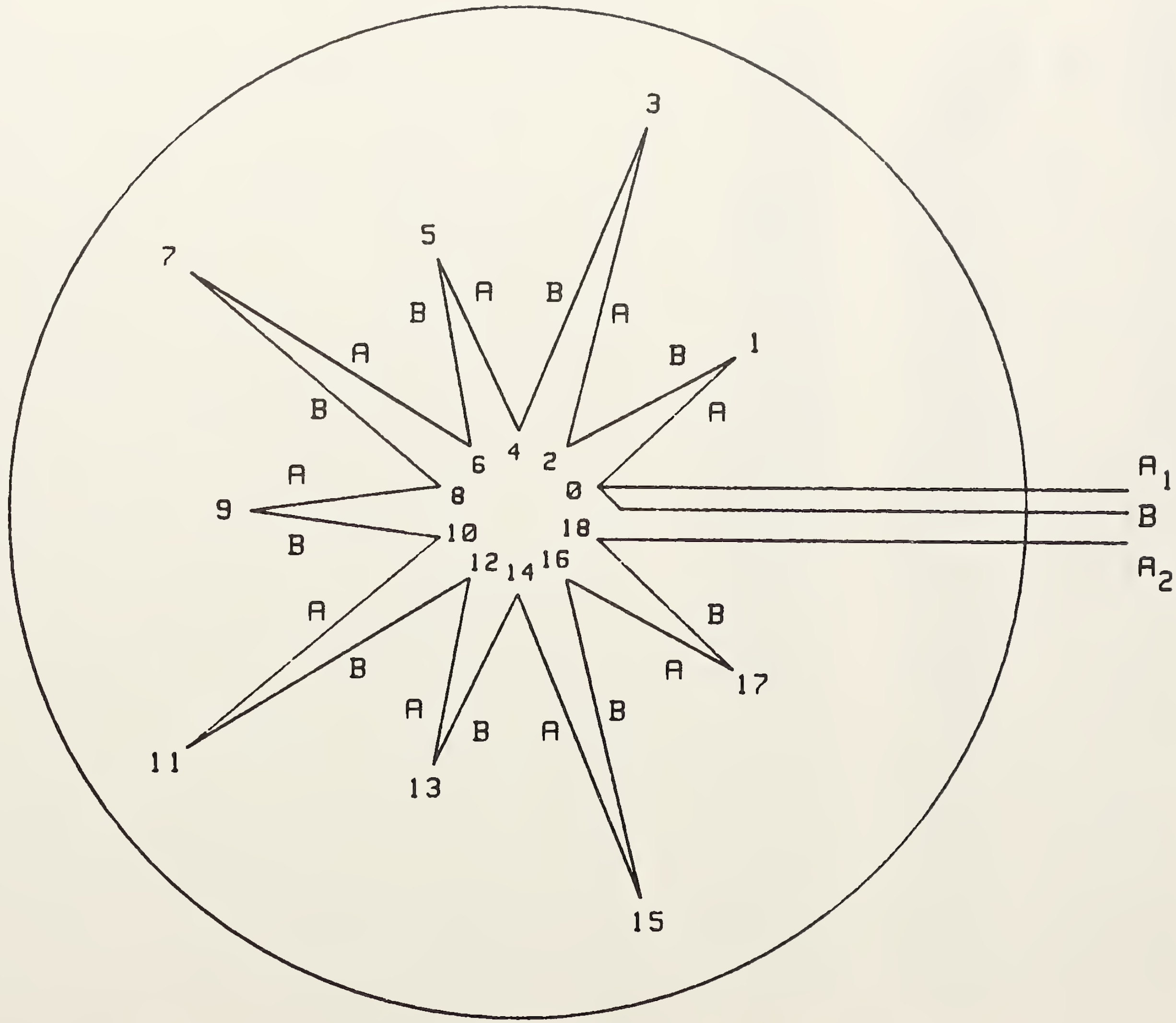

Figure 1. A thermocouple-based device for the determination of average surface temperature. Leads A_1 and A_2 are of the same material as A in the thermocouple.

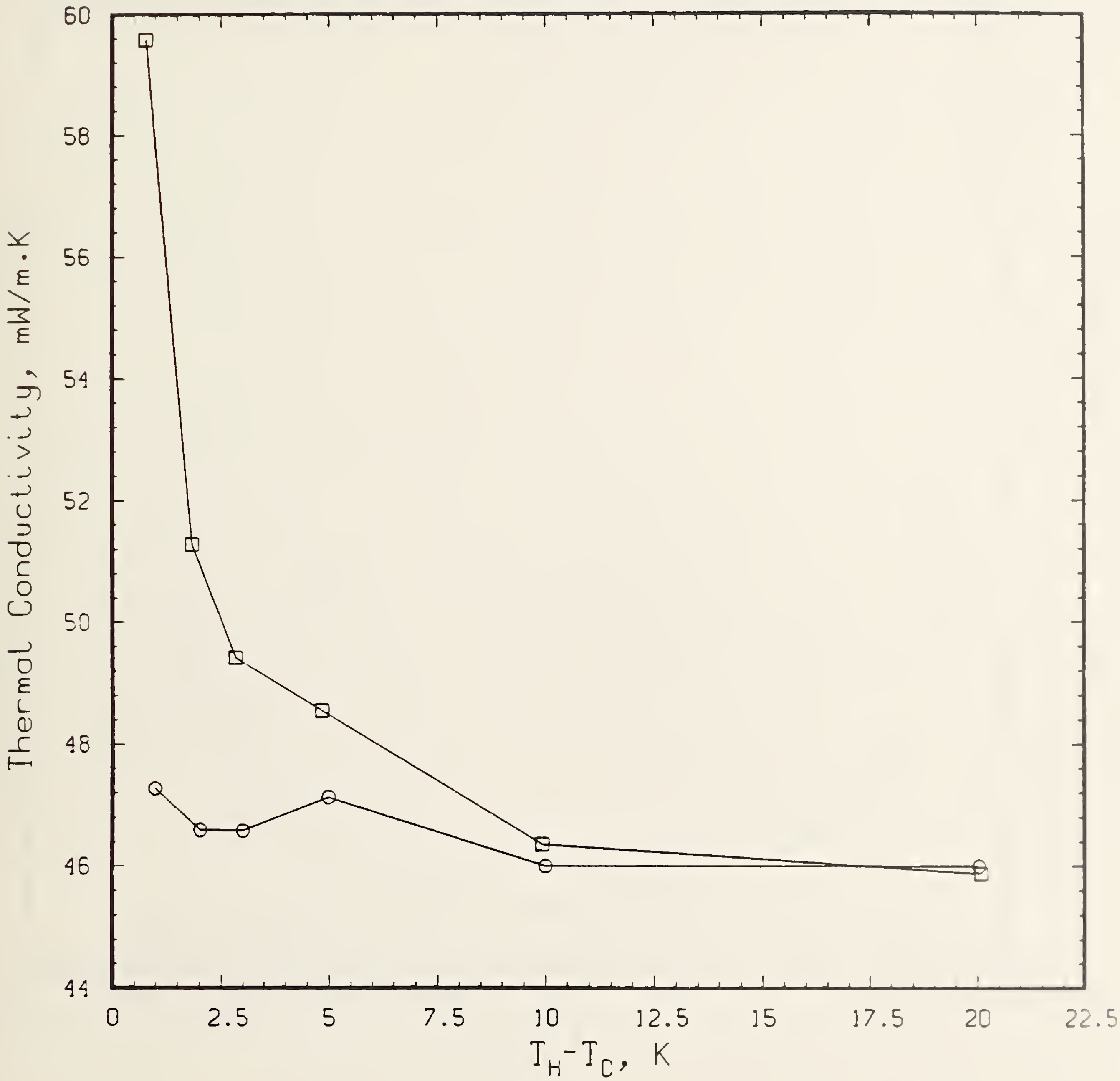

Figure 2. Thermal conductivity of an insulating specimen versus temperature difference as determined by the average surface temperatures (circles) and the single thermocouple on each surface (squares) for the double-sided mode of operation.

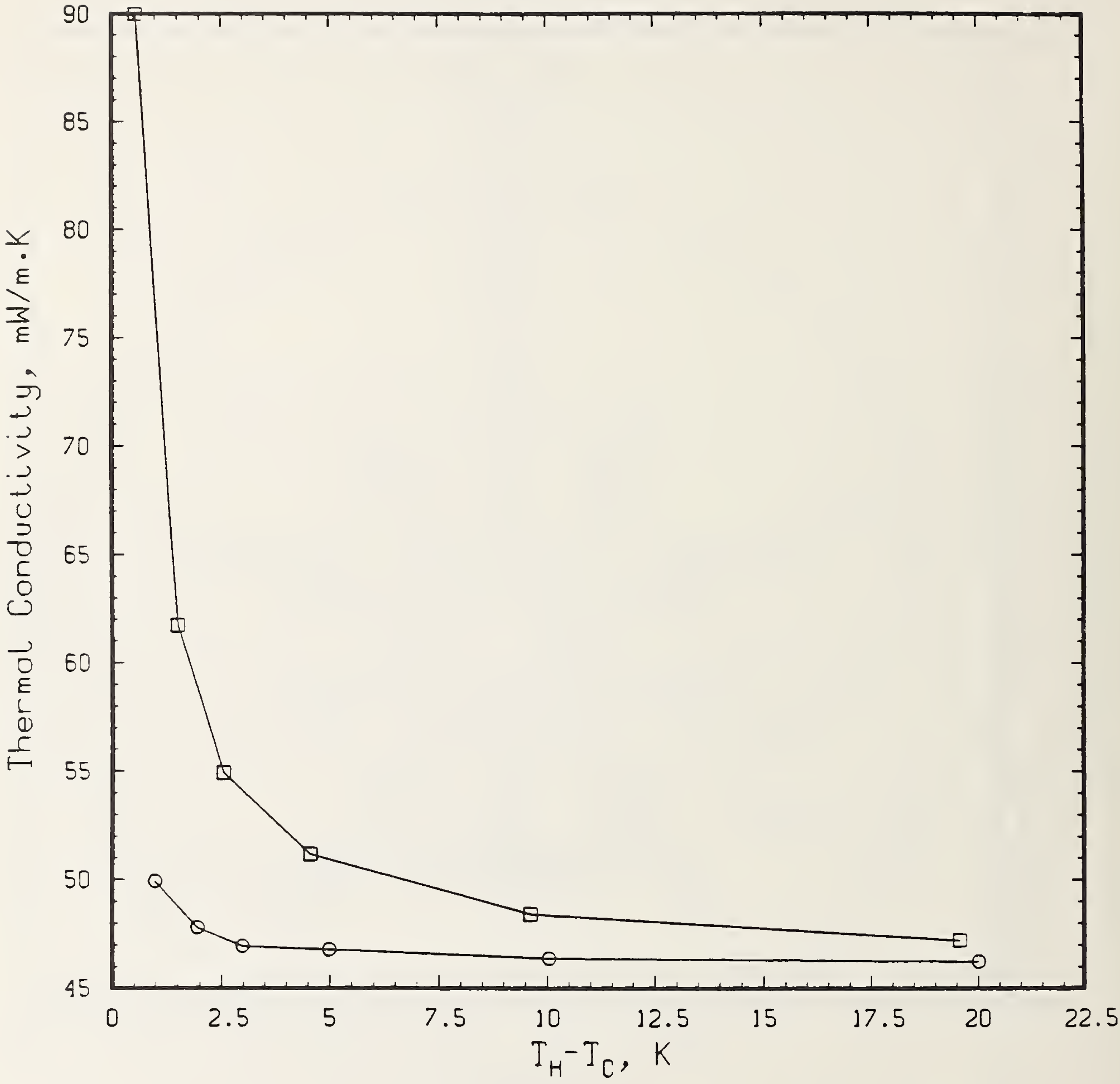

Figure 3. Thermal conductivity of an insulating specimen versus temperature difference as determined by the average surface temperature (circles) and the single thermocouple on each surface (squares) for the upper specimen in the single-sided mode of operation.

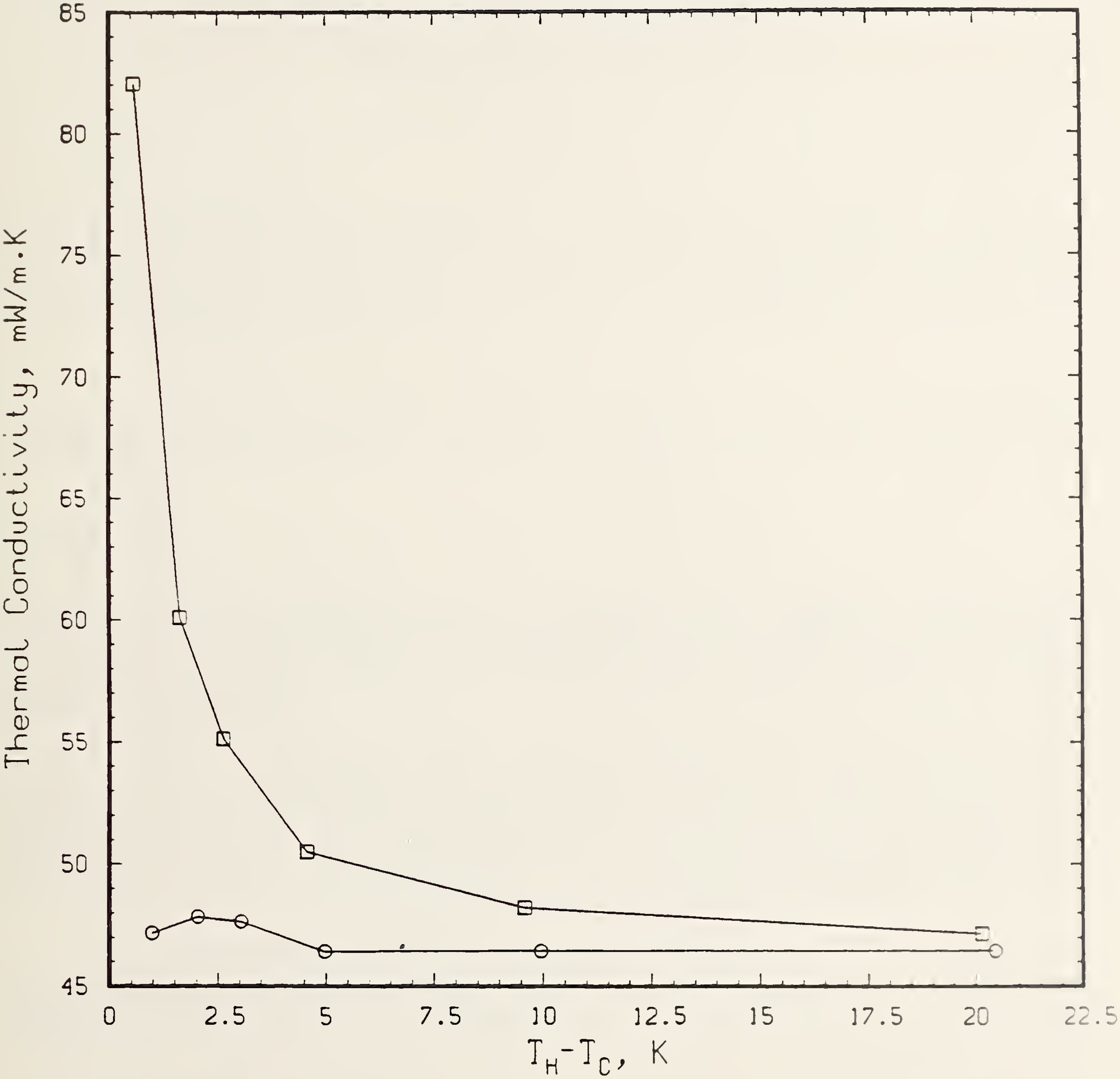

Figure 4. Thermal conductivity of an insulating specimen versus temperature difference as determined by the average surface temperature (circles) and the single thermocouple on each surface (squares) for the lower specimen in the single-sided mode of operation.

Appendix C: A Modified Digital PID Temperature Controller for Thermal Properties Measurements*

Jerome G. Hust, B. James Filla, and David R. Smith
Chemical Engineering Science Division
National Bureau of Standards
Boulder, Colorado

Abstract

A modified digital PID temperature controller for thermal properties measurements is described. Data are presented to illustrate the approach to equilibrium and the degree of stability at equilibrium. The controller is being used to control the heating elements of a high temperature guarded hot plate to within approximately 10 mK.

Key words: derivative control, digital computer control, integral control, proportional control, thermal properties, temperature controller, digital PID controller.

INTRODUCTION

Thermal properties testing often requires a high degree of temperature stability over long periods of time to achieve accurate results. For example, steady-state thermal conductivity guarded-hot-plate and heat-flow-meter apparatus require stabilities in the 10 mK range to produce accuracies on the order of one percent.

In the past these stabilities were obtained either by waiting long periods of time with constant power input or with analog controllers. With the advent of computerized automation, digital controllers were introduced. Digital computer controllers allow considerably more flexibility in the design of controller logic, because software can be used to perform complicated functions that are virtually impossible to accomplish with hardware.

This paper describes a proportional-integral-derivative (PID) controller that is being used to control the temperature of the heating elements in a recently completed high temperature guarded-hot-plate apparatus [1] at the National Bureau of Standards, Boulder, Colorado.

GENERAL

The basic equation for a PID controller is

$$m=m_o + K(e+ \frac{1}{\tau_I} \int_o^t edt+ \tau_D \, de/dt) \tag{1}$$

where e is the controller input error signal, m is the controller output signal, m_o is the controller bias, t is time, K is the proportional gain, τ_I is the integrator time constant, and τ_D is the differentiator time constant. The latter three parameters are usually considered to be constants set by the operator to optimize the controller operation for a specific application.

In a digital computer control system the input error signal is sampled at discrete times rather than continuously as in an analog system. To accommodate discretely sampled error signals eq (1) is replaced by the approximation,

$$m_i = K(e_i+\frac{1}{\tau_I} \sum_{j=1}^{i} t_c e_j + \tau_D(e_i-e_{i-1})/t_c) \tag{2}$$

where t_c is the time between sampled error signals and the subscript i denotes the current reading.

In addition, the operation of the controller can be improved by making the control parameters dependent on the magnitude of the input error signal. For example, with respect to the proportional gain, it is desirable to have a high value of gain when the error signal is large and a smaller value of gain near the set point. This allows a rapid approach to equilibrium with robust control. Alternatively, the proportional control becomes more quiescent near the setpoint, stabilizing the controlled output variable, m. This feature is especially important on control loops where both the error signal fluctuation and the controller output fluctuation must be minimized.

An approach opposite to that used for proportional gain adjustment is taken with respect to the integral gain adjustment, where integral gain is defined as $1/\tau_I$. If a fixed integral gain adjustment, optimized near the setpoint, is used in a digital controller it will work well near the setpoint. However, a large positive setpoint change will result in a large error signal and rapid growth of the integrator sum. This reset windup can cause severe setpoint overshoot and oscillation. The adverse effects of reset windup can be minimized if integral gain is very small for large values of e_i. As the setpoint is approached, the value of integral gain is increased until it reaches an optimum value. This run-time adjustment of control parameters results in controller operation based primarily on proportional control when e_i is large and on integral control when e_i is small.

The time required to achieve system stability can be reduced through the addition of derivative control so long as the error signal is relatively free of noise. Using this criterion, one can argue that derivative gain should be zero when e_i is within the noise band of the system and adjusted to some optimized value when the error signal is sufficiently larger than the noise band. This allows the derivative control to operate normally when the system is approaching equilibrium, but prevents it from magnifying noise in the error signal during steady-state operation at the setpoint. As with each of the other modifications, this allows the controller to operate with optimum parameter settings both during the approach and at achievement of system equilibrium. In a software-based control system such behavior is easily incorporated into the control logic. These concepts, when incorporated into eq (2), result in eq (3),

$$m_i = KK_pe_i + \sum_{j=1}^{i} t_ce_jKK_I/\tau_I + KK_D\tau_D(e_i-e_{i-1})/t_c \tag{3}$$

The digital controller used in the NBS guarded hot plate is based on eq (3)with the following functional forms for K_P, K_I and K_D:

$$K_p = (1-(1-K_{P1})\exp(-(ei/S_{p1})^2))(1-(1-K_{p2})\exp(-(e_i/S_{p2})^2)) \tag{4}$$

$$K_I = \exp(-(e_i/S_I)^2) \tag{5}$$

$$K_D = -(1- \exp(-e_i/S_D)^2)) \tag{6}$$

where K_{P1} K_{P2}, S_{P1}, S_{P2}, S_I, and S_D are constants, along with K, τ_I, and τ_D, to be determined by optimizing the controller operation.

These functional forms are illustrated in Figure 1.

OPTIMIZATION OF THE PARAMETERS

The optimization procedure for a control system involving a large number of parameters can be frustrating unless a systematic approach is used. The following approach has been found to simplify the optimization task:

A. Using proportional control only, with K_P set equal to unity, find

1. the maximum value of K, K_{min}, that produces no overshoot in the controlled variable. Also determine the stable value of the error signal, e_o, at this gain.

2. the value of K, K_{max}, that produces constant amplitude oscillations. Also determine the period of these oscillations, τ_p.

3. the standard deviation of the error signal, σ_n, at stability without feedback control other than a bias to maintain stability.

B. Acceptable adjustment of the PID control can be obtained as follows:

1. Set S_{p2} and S_D equal to four times the standard deviation σ_n, and S_I to $2e_o$.

2. Set K to $K_{max}/2$, K_{P1} to K_{min}/K, and K_{P2} to 0.95.

3. Set τ_I to 1.2 τ_P.

4. Set τ_D to produce an acceptable rate of change of the controlled variable at the maximum error signal.

5. Set S_{P1} such that the approach to the setpoint is as rapid as possible while maintaining acceptable overshoot.

RESULTS

To illustrate the characteristics of the modified digital controller, results are presented as obtained from the bottom auxiliary heater of the NBS high temperature guarded-hot-plate apparatus. This plate is constructed of alumina with a thickness of 2.5 cm, a diameter of 25.4 cm, and a total mass of 3.2 kg. The heater element is powered by a programmable 275 watt direct current power supply. The control sensor, mounted on the outer edge of the heater plate, is a platinum resistance thermometer with an ice point resistance of 100 ohms. The thermometer current ranges from 0.5 to 1.5 milliamp depending on the set point temperature. The sensor voltage and current are determined with a 5 1/2 digit DVM.

Figure 2 shows the initial approach to the setpoint, 356.15 K, for a large initial offset signal. Note the rapid approach to equilibrium without significant oscillation on the scale of this graph. Figure 3 shows the control band as final equilibrium conditions are approached.

Automation of the apparatus using this control algorithm has eliminated the need for time-intensive operator interaction with the instrument. Concurrently, system control is achieved that equals or exceeds the most sensitive manual operation presently obtainable with this apparatus. Because of the high degree of adaptability and the resulting robust yet delicate control obtained with this algorithm we feel that it could be quite useful in a variety of digital computer control applications.

REFERENCES

1. Hust, Jerome G.; Filla, B. James; Hurley, James A.; and Smith, David R., An automated High Temperature Guarded-Hot-Plate Apparatus for Measuring Thermal Conductivity of Insulation Between 300 and 800 K, to be published as a National Bureau of Standards Report.

List of Figures

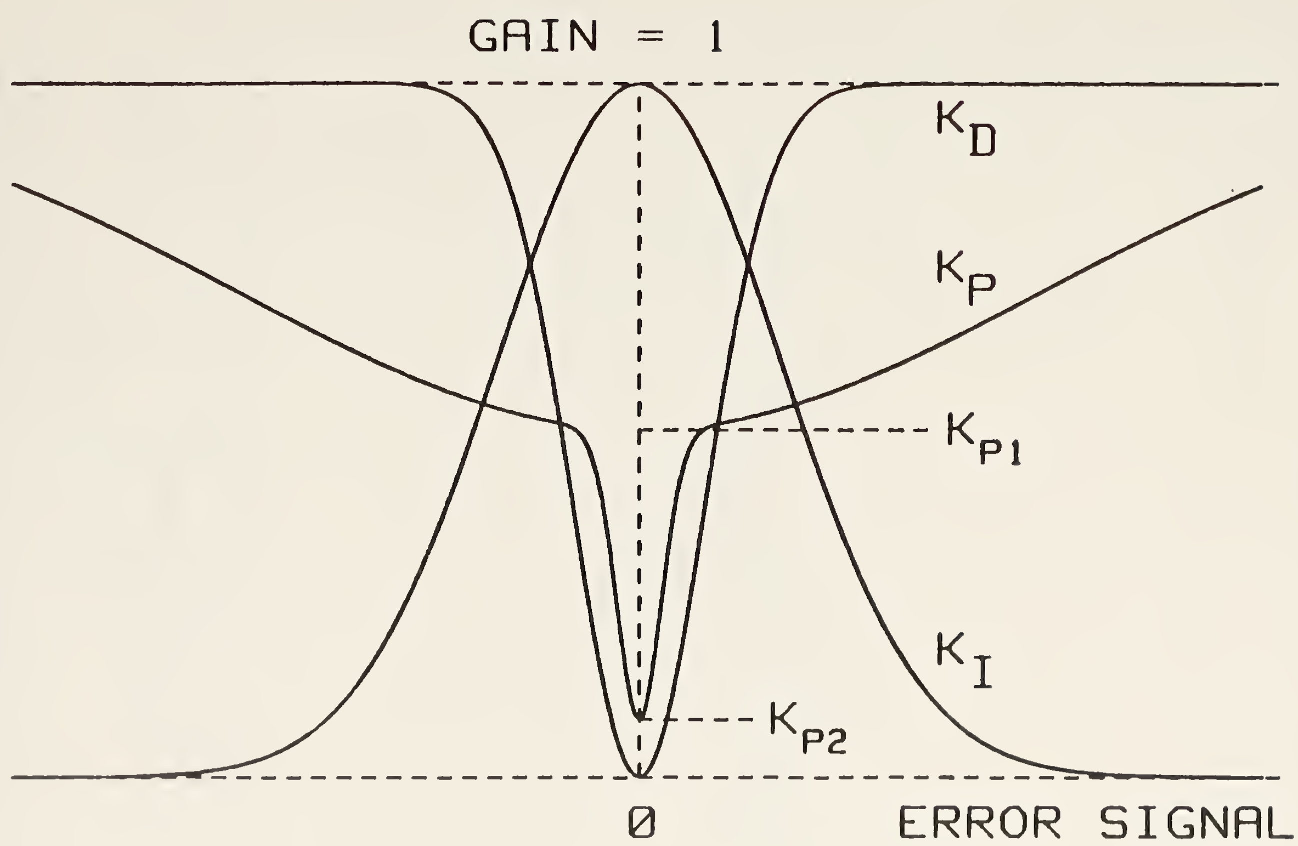

Figure 1. Graphical representation of the functional forms K_p, K_I and K_D.

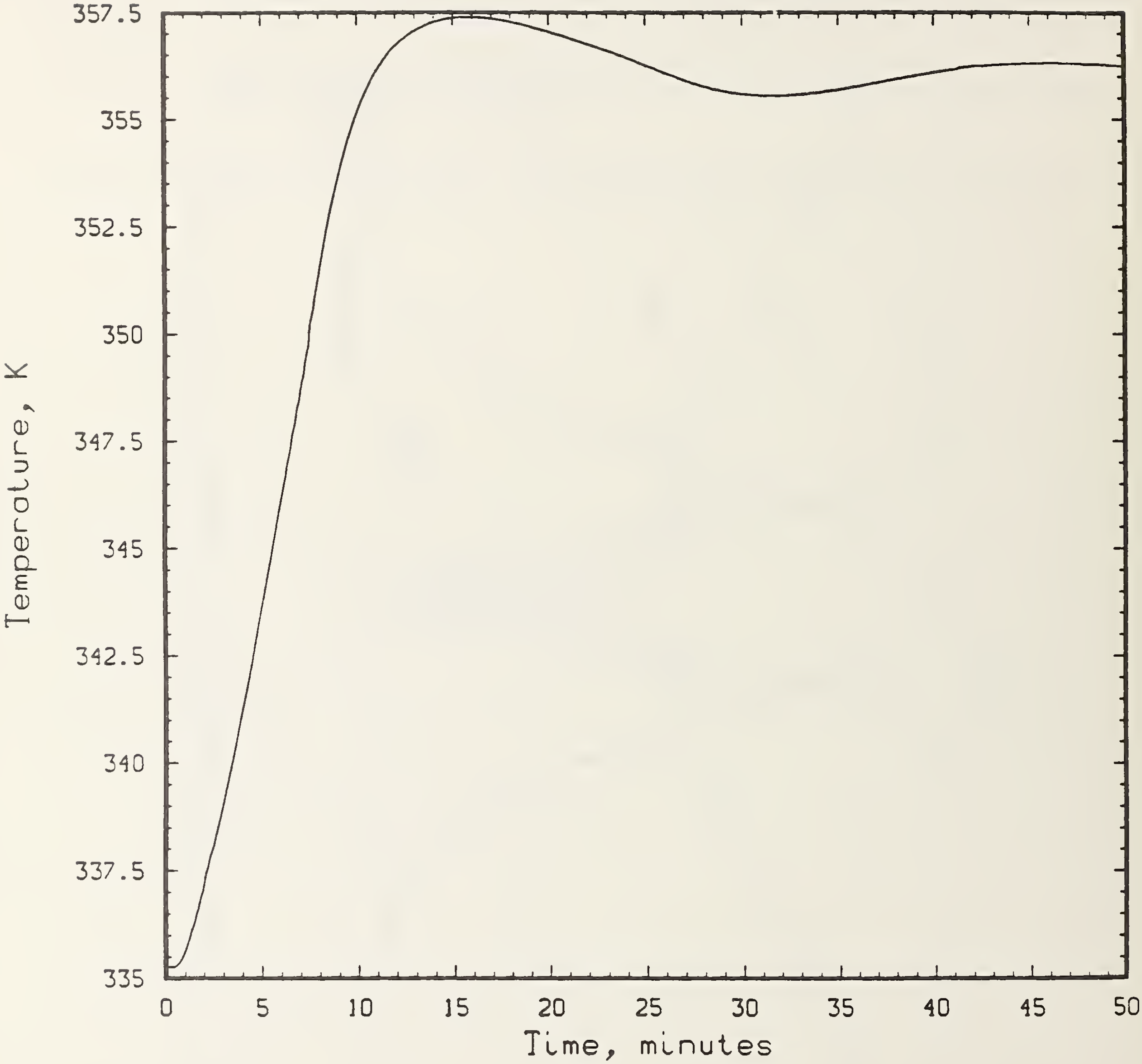

Figure 2. Typical initial approach to equilibrium after a step setpoint change for the bottom auxiliary plate of the NBS high temperature guarded hot plate apparatus.

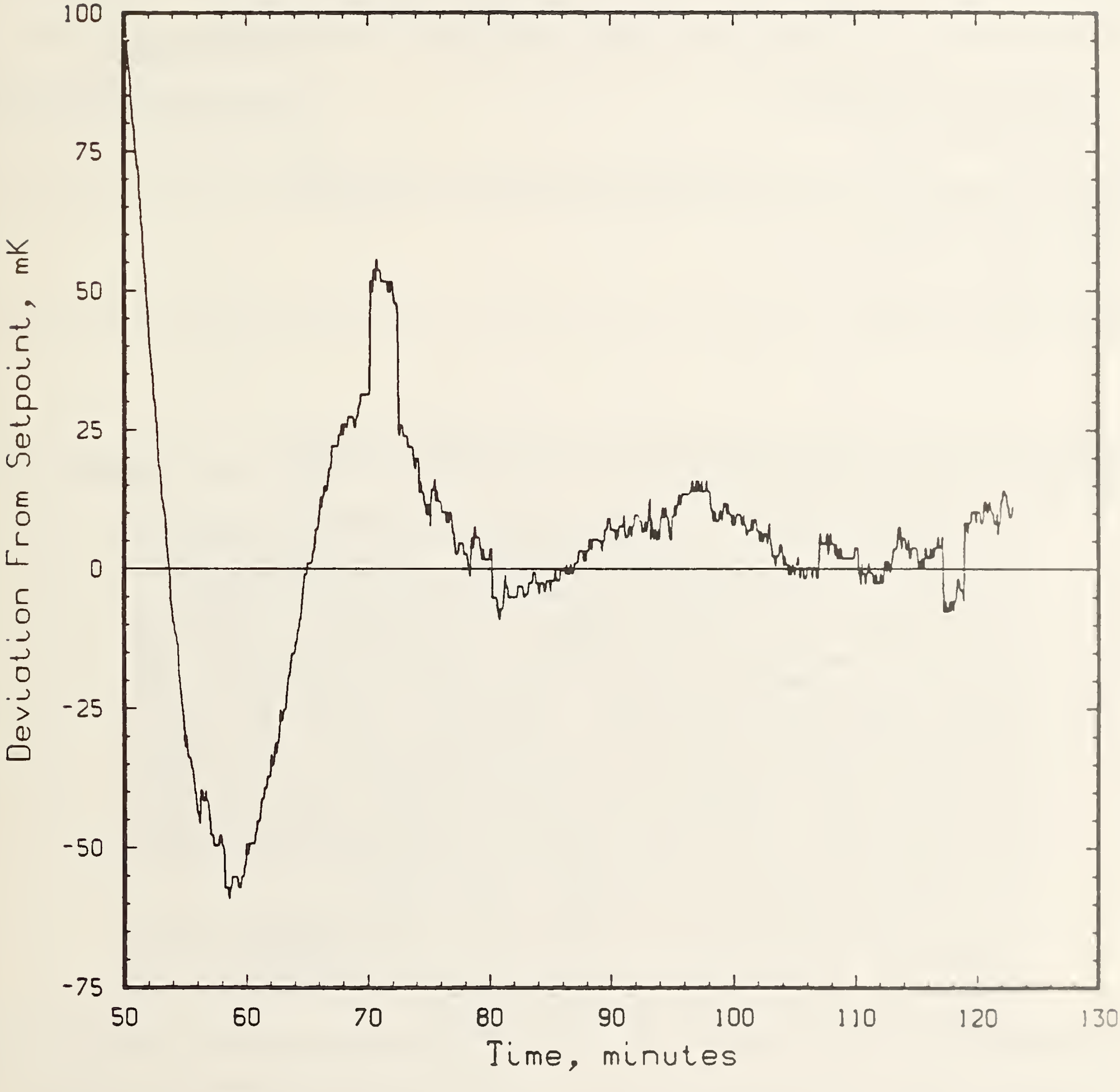

Figure 3. Typical deviations from the set point following the initial approach to equilibrium for the bottom auxiliary plate.

NBS-114A (REV., 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 88-3089	2. Performing Organ. Report No.	3. Publication Date May 1988

4. TITLE AND SUBTITLE

AN AUTOMATED HIGH-TEMPERATURE GUARDED-HOT-PLATE APPARATUS FOR MEASURING APPARENT THERMAL CONDUCTIVITY

5. AUTHOR(S)

Jerome G. Hust, B. James Filla, James A. Hurley, and David R. Smith

6. PERFORMING ORGANIZATION (*If joint or other than NBS, see instructions*)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (*Street, City, State, ZIP*)

Department of Energy
Oak Ridge Operations
P.O. Box E
Oak Ridge, TN 37831

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (*A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here*)

An automated guarded-hot-plate apparatus was designed and built to meet the requirements of ASTM Standard Test Method C-177 for measuring the thermal conductance of thermal insulation. Apparent thermal conductivity has been measured with this apparatus in the range from 40 to 100 mW/(m.K) at mean temperatures from 300 to 750 K, in environments of air and helium, at pressures ranging from atmospheric pressure to roughing-pump vacuum. The apparatus is controlled by a desk-top computer. A thermocouple device of novel design more accurately senses the average temperature over the surface of each heater plate. An improved algorithm for the control sequence leads to more stable heater powers and specimen temperatures. Initially it brings the system rapidly to a temperature setpoint with minimal overshoot. It also permits highly sensitive control of the plate temperatures in later phases of the measurement sequence when thermal stability of the specimen boundaries is very important in measuring the thermal conductivity with high precision. This algorithm has enhanced the performance of both the high-temperature and the low-temperature guarded-hot-plate apparatus at this laboratory. The apparatus can be operated at either constant hot-plate temperature or constant heater power.

Overall uncertainties of apparent thermal conductivities at atmospheric pressure are 2% at 300 K and 5% at 750 K when measuring conductivities in the range from 40 to 100 mW/(m.K). The apparatus will be valuable in development of new Standard Reference Materials of low conductivity and for higher temperature ranges, and is being used in comparative interlaboratory measurement programs.

12. KEY WORDS (*Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons*)

automated control system; guarded-hot-plate apparatus; high temperature; mean-temperature sensor; thermal conductivity; thermal insulation; thermocouple device

13. AVAILABILITY

☐ Unlimited
☐ For Official Distribution. Do Not Release to NTIS
☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

134

15. Price

USCOMM-DC 6043-P80